Department of Computer science

University of Gujrat

Title:

"KFM Pharmacy"

LOGO

Session: BSCS _2020

Project Advisor: Ma'am Saliha Zahoor

Submitted By:

Muhammad Munir	20021519-008
Khurram Shahzad	20021519-036
Muhammad Faizan	20021519-115

FACULTY OF COMPUTING AND INFORMATION TECHNOLOG

Acknowledgement

We truly acknowledge the cooperation and help make by **Miss. Saliha Zahoor** Assistant Professor , University of Gujrat. She has been a constant source of guidance throughout the course of this project. We are also thankful to our friends and families whose silent support led us to complete our project.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF GUJRAT

Table of Contents

CHAPTER 2 SYSTEM REQUIREMENT SPECIFICATIONERROR! BOOKMARK NOT DEFINED.

1.1.INTRODUCTION..... 5

1.1.2. REQUIREMENTS SPECIFICATION..... 5
1.1.3. SYSTEM MODELING 5
1.1.4. REQUIREMENTS VALIDATION 5
1.1.5. REQUIREMENTS MANAGEMENT 5
1.1.6. IDENTIFY EXTERNAL INTERFACE 6
1.1.7. DEVELOPMENT OF CONTEXT DIAGRAM 6
SYSTEM TO BE DESIGNED IN A PROJECT. IT POINTS OUT THE FLOW OF INFORMATION BETWEEN THE SYSTEM AND EXTERNAL COMPONENTS..... 7
1.1.8. CAPTURE "SHALL'' STATEMENTS 7
1.1.9. ALLOCATE REQUIREMENTS 7
1.1.10. PRIORITIZE REQUIREMENTS..... 7

2.1. SYSTEMS SPECIFICATIONS..... 7

2.1.2. INTRODUCTION..... 7
2.1.3. EXISTING SYSTEM..... 8
2.1.4. ORGANIZATIONAL CHART..... 9
2.1.5. FUNCTIONAL REQUIREMENTS 10
2.1.6. NON-FUNCTIONAL REQUIREMENTS 14
2.1.7. IDENTIFYING EXTERNAL ENTITIES 16
2.1.8. CLASS DIAGRAM 18
2.1.9. USE CASE DIAGRAM..... 19

3.1. INITIAL REQUIREMENTS FOR ALL CLASSES 24

3.1.1. REQUIREMENTS FOR THE ADMIN CLASS: 24
3.1.2. INITIAL REQUIREMENTS FOR THE MANAGER CLASS:..... 25
3.1.3. INITIAL REQUIREMENTS FOR THE EMPLOYEE CLASS:..... 26
3.1.4. INITIAL REQUIREMENTS FOR THE CLIENTS CLASS: 27
3.1.5. INITIAL REQUIREMENTS FOR THE DOCTOR CLASS:..... 28
3.1.6. INITIAL REQUIREMENTS FOR THE SALESMEN CLASS: 29
3.1.7. INITIAL REQUIREMENTS FOR THE INVENTORY CLASS:..... 30
3.1.8. INITIAL REQUIREMENTS FOR TECHNICAL STAFF: 31

3.2. ALLOCATE REQUIREMENTS:..... 32

3.3. PRIORITIZE THE REQUIREMENTS 41

3.4. TRACEABILITY REQUIREMENT MATRIX 49

3.5.SEQUENCE DIAGRAM..... 59

4.1. INTRODUCTION: 60

4.1.2. DOMAIN MODEL..... 60
4.1.3. WHAT IS DOMAIN MODELING?..... 62

4.1.4. SYSTEM SEQUENCE DIAGRAM 62
4.1.5. SEQUENCE DIAGRAM 62
4.1.6. COLLABORATION DIAGRAM 64
4.1.7. OPERATION CONTRACTS 65
***4.1.8.* DESIGN CLASS DIAGRAM 69**
***4.1.9.* STATE CHART DIAGRAM 70**
***4.1.10.* DATA MODEL 71**

1.1.Introduction

Requirements engineering process provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system. The task of capturing, structuring, and accurately representing the user's requirements so that they can be correctly embodied in systems which meet those requirements (i.e. are of good quality).

1.1.2. Requirements specification

Requirement specification, also known as documentation, is a process of jotting down all the system and user requirements in the form of a document. These requirements must be clear, complete, comprehensive, and consistent.

1.1.3. System modeling

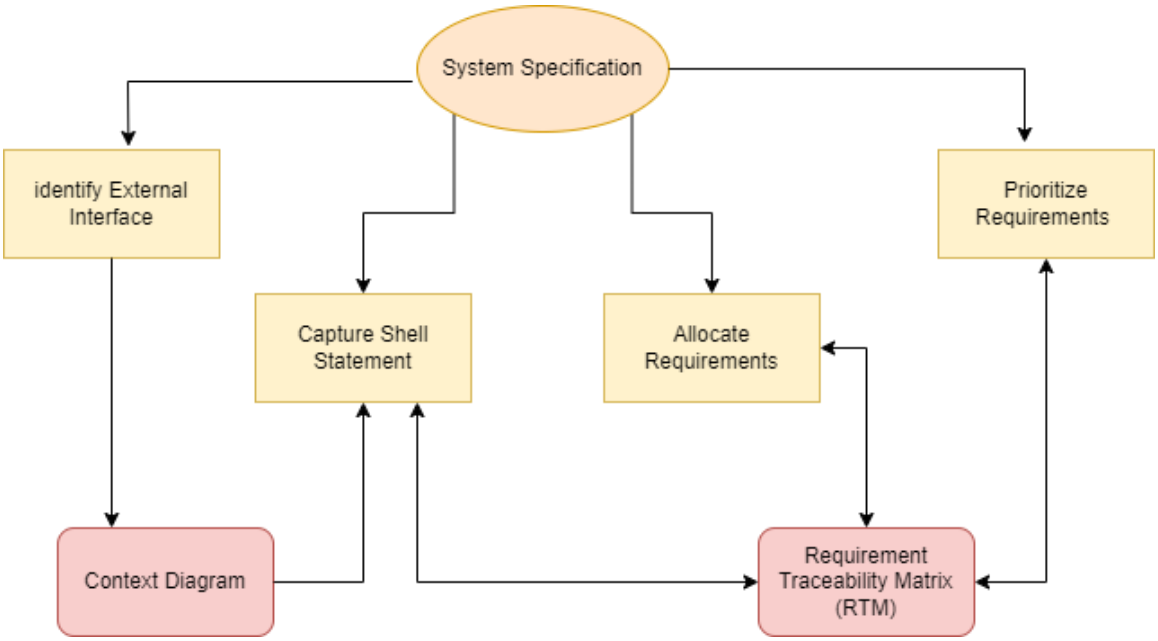
is the process of developing abstract models of a system with model presenting a different view or perspective of that system.

1.1.4. Requirements validation

Requirements validation is the process of checking that requirements are defined for development and defining the system that the customer really wants. To check issues related to requirements, we perform requirements validation. Requirements validation usually helps us detect errors at an early stage of product development so that it does not result in excessive rework when detected later in the software development life cycle. In this blog post, we will look at what Requirements Validation is, its Process, and various Tools used for Requirements Validation.

1.1.5. Requirements management

Requirements management (sometimes called requirements engineering or requirements definition) is among the first steps of a Product Lifecycle Management process. It's how companies take raw ideas into more detailed requirements. Requirements management involves documenting, analyzing, prioritizing, and visualizing what stakeholders want from your products and services. Requirements management also typically includes change management for requirements, requirements traceability, collaboration with relevant stakeholders, and approvals of requirements.



Here, requirements specification is to be discussed. Requirements specification would lead to the following four steps

1.1.6. Identify external interface

In this field we have to tell the about the external interface and the way of working, that if one user enters to system that how our system will interact with the user and how it will the give the look, how it will give a look to user and how the interaction between them will take the place, In external interface it will be a catchy option for the users to keep them resist on the website.

In this field we also include the interaction of hardware and software for the software, so the system can interact with the not only components and but also with the users in the proper way.

1.1.7. Development of context diagram

the **Context diagram** is the highest level in a Data Flow Diagram. It is a tool popular among Business Analysts who use it to understand the details and boundaries of the

system to be designed in a project. It points out the flow of information between the system and external components.

1.1.8. Capture "shall" statements

In "Shall" statement we capture the functional requirements that will be used to record the requirements for the software, these requirements will be used to understand the basic function of software and the team will use this shall requirements to develop a complete code for the development of software.

1.1.9. Allocate requirements

Requirements Allocation is the act of decomposing higher-level requirements and assigning them to lower-level functions. All requirements of the top-level functions must be met by the aggregate of those for all lower-level functions.

1.1.10. Prioritize requirements

Requirements prioritization is essential to any well-run project. It ensures that the project focuses on the most important elements first, and that everyone understands and agrees regarding what the project's most important elements are. Good prioritization of requirements will also ensure that engineers, programmers and database analysts develop a project's most critical elements in sync with the business needs.

Development of requirements traceability matrix

2.1. Systems Specifications

The following are the clauses that must be included while describing the system specifications.

2.1.2. Introduction

Information and technology are growing rapidly with every day and making the life easier for all those who are familiar and able to use the technology. The current era is for those who are using the technology tools every day and all the people nowadays are using the technology in one or in another way to make the time use effective and to improve the productivity. All those countries and people far succeed than those who are reluctant to use the technology and do not adopt the new way of life in the current world.

Now with the advancement of technology, the use of computers and the internet is at its height, so those who are on the internet and using the computer are getting a lot of success and progress. As the computer has been involved in every activity of life so it was also helpful In the field of Health and care.

To exploit the ICT in the health care system, a Pharmacy management system is being built, Pharmacy management system is robust, integrated technology. The pharmacy management system deals with the maintenance of drugs and consumables in the pharmacy unit. The set-up of this pharmacy management system will ensure the availability of a sufficient quantity of drugs and consumable materials for the patient. This will enhance the efficiency of clinical work and ease patients' convenience, bearing in mind that Ethiopia is heading towards pharmaceutical care of patients. In addition, the pharmacy management system will be able to process drug prescriptions with ease. PMS will design to detect drug interaction.

In general, the pharmacy management system is based on computer technology that gives service for users, managed by the pharmacist who gives an implementation of function relatively ineffective times as well as will design for removing time-wasting, saving resources, easy data access of the medicine, security on data input and data access by removing almost manual based system.

A pharmacy management system is software that will be used to make the work more efficient and accurate in the field of medicine and it will also improve the work efficiency of the pharmacy and be helpful to the owner and also the user to manage their chaos in a perfect and ease of use. This software will be helpful for not only the client/owner to manage his business pharmacy effectively but also will give great exposure and use to users, doctors, staff, and also for the person who will manage the system.

Here in this project, different types of actions will be performed, which will be effective and useful for all the stakeholders.

2.1.3. Existing System

With the help of the right software, you will be able to deliver the highest level of care to your patients while it will also help in maximizing profits and allow you to spend more time generating new business. There are many such benefits associated with

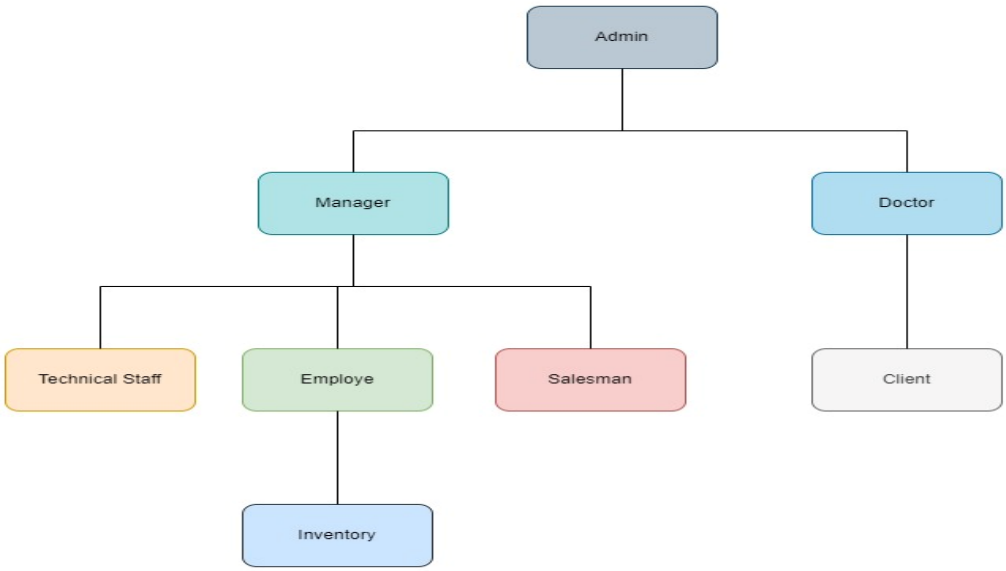
Pharmacy Management Software, so let's explore the advantages of pharmacy management system:

* Low cost of ownership

* It is helpful in increasing productivity and resultantly profitability.
* It is a flexible system that can be customized as per your requirements.
* There are wide-ranging modules created keeping in mind all the requirements, with a separate login for pharmacy administrators &employees.
* With it being integrated with Barcode Scanner, it is able to record details easily and hence speeds up the sale procedure
* lt is a lot more than just a Pharmacy Management System, as it also takes care of Supply Chain Management
* There are a number of useful features available in the system such as Expiry Date Alert & Minimum Stock Alert
* Quite promptly & quickly the system provides the analysis and information about the business, which is helpful in better decision-making
* It offers a Centralized Data Storage facility as well as Easy & Interactive retrieval of data.
* The system is reliable & secure as during the development of the system, all the major threats were considered and all the preventions.
measures are also taken care of.
* lt boats a simple, user-friendly & efficient user-interface design the key elements are taken care of during the designing of the system.

2.1.4. Organizational Chart

Organizational chart will be very much supportive to get a better overview of the organization's business areas and their decomposition into different departments.



Scope of the System

The scope of this project is limited to the activities of a pharmaceutical store which includes will improving health outcomes, reduce hospital and long-term care admissions, enhance access and care in the Estate and surrounding communities and ensuring best use of resources, the use of a computer based management system for improving the efficiency of a pharmacy is needed and it is an essential part of any modern continuously evolving society. The system will not be able to handle drug prescription, drug to drug interaction. The system will not be able to handle contraindication and poly pharmacy in a prescription; this implies that these services will be manually completed by the pharmacist.

Summary of Requirements (Initial Requirements)

we can develop such system only on the demand and requirement of system's users. So, now we will start to take to take the requirement by the pharmacy's owner/stakeholders and then after getting the complete requirement briefing from the client we will move to further steps.

2.1.5. Functional Requirements

1. Admin functions:
    1) Admin will be a authentic user and will be a person who can access the system with login.
    2) In this system there will be a Username and Password for the Admin.
    3) The admin can be a only one single person in system, more than one admins will not be allowed by the system.
    4) New admin can be added with the permission of previous admin.

5) New admin login will logout the first admin login.
6) Admin will be able to add the employee in system.
7) To delete the record of specific employee of system.
8) To update the record in any field.
9) Admin will be able to make the link with doctors.
10) Admin can communicate to salesmen directly without any problem and can generate reports.
11) Admin should be able to check the performance of all branches and can generate the finance reports form the system.
12) Reports should be generated daily, weekly, monthly and on yearly basis.

Manager:

1) There will be a login for the manager to use the system.
2) Manager will have the unique Username and password.
3) The manager can add the employee.
4) Manager credentials consist of name, password, address, phone_number, unique_id.
5) He can delete the record of employee (But data must be saved in file for the admin)
6) He can update record (like salary, working hours, location etc.)
7) Manager can check the inventory.
8) Manager can make the link with the salesmen directly (but this link will be recorded for admin).
9) Manager should be able to check the record of more than one pharmacy.
10) In any change by the manager, the activity of manager will be saved with his credentials.

System Users:

1) The user for the system must login first to use the system.
2) For just login we will need of name/email and password.

3) But if user buy or order the medicine then should be capable of recording the other credentials (name, password, address, phone_number).
4) There will be a unique id for each user of system.
5) The id will consists on Username and password.
6) The user can also join the system using the google system or any other official website.
7) If user order the medicine then there should be a option for the credit card, debit card and cash on delivery.
8) After order approval, user must be able to track his parcel.
9) There should be option to cancel the order within the twenty-four (24) hours, after that should be no option to cancel the order.

Staff/Employee:

1) The staff must have a login.
2) Login will be consisted of username and password.
3) The staff will be able to check the inventory.
4) Staff can update the state of inventory (if staff change inventory, he must login first and all the activities will be saved with his name and password and with particular period of time).
5) Staff name, password, residential address, phone number, working address and unique will be recorded in the system.
6) Staff can check the salary status.
7) Staff can generate the receipts for the user of system.
8) Staff can't change name or password.
9) Staff can send the report to manager for particular activity.
10) Staff can make the changes to any part of inventory (name, price, quantity, location of inventory).

Salesmen:

1) Only authentic salesmen can login for the system.

2) Each salesman consists of name and password.
3) Salesmen can check the inventory of pharmacy and it's branches.
4) In any sale, salesmen can request for his commission to manager.
5) Salesmen can generate a report for his sales to keep the record.
6) Salesman can view the data and can't make any minor change in the system.
7) Salesmen name, password, address, phone_number and also working location will be recorded in the system.
8) Salesmen can check his balance due to any linked pharmacy.
9) He can generates the money trail for his income.
10) Salesmen performance for sales will be recorded and will be awarded with the grade highest to lowest.
11) Salesmen can download the pictures of medicines with his account for use.

Inventory:

1) Inventory will get a unique_id in system.
2) For each inventory the date must be recorded.
3) For each medicine name, price, source, and destination will be recorded in the system.
4) Any change to inventory can be done by the staff or by the manager, and in such situation a report will be generated.
5) System will keep the record of medicines.
6) Staff can mark the medicine as sold out.
7) User can keep the medicine in wishlist or in add to cart.
8) There will be option to show the discount on specific medicine.
9) Sold medicine should be recorded in separate file with buyer id and it will be used for future decisions.
10) The most and less sold items can be sorted at the time of need.
11) Receipts for the medicines can be generated.
12) Notification must be given by the system before of one week to end of inventory for the specific item.
13) Inventory items should be visible through the website.

14) Picture must be uploaded for item.
15) User must be able to order medicines as according to his requirement.

Doctors:

The doctors will also be associated with the pharmacy and its interrelated branches.

1) The doctor can generate his account with approval of admin.
2) To use functions of system, doctor must have to login.
3) The login will consist of unique id, name, and password.
4) The data about the doctor in system will consists of unique id, name, password, residential address, clinical address, his specific field in medical.
5) The doctor can prescribe the medicine and report can be generated through the system.
6) The doctor can check the history of the patient's usage of medicine through the system.
7) If a doctor refers patients to pharmacy, then he can earns a commission and can get paid for it.
8) A doctor can download his receipts for income through the pharmacy.
9) A doctor can see his referral history.
10) A doctor can delete his account from the system.
11) A doctor can update his credentials with the approval of the admin.

Technical staff:

1) Must create the login for use.
2) Credentials for the system (name, password, address, designation, role).
3) Can access the code.
4) Maintenance of system.

2.1.6. Non-functional requirements

• User-Friendly Record System: The purpose of the pharmacy management system is to input & recall patient, drug & prescription information easily. This system is quite user-friendly, and it has

the ability to keep a record of patient, drug & prescription information, which is certainly come useful for future purchases or other purposes.

• HIPAA and Regulatory Compliance: It goes without saying that the pharmacy system must be aligned with legal requirements in the field hence it must comply with HIPAA & regulatory compliance (consisting of FDA-required Med Guides & DEA Reporting.

• Full Prescription Management: As integrated with ePrescription, the system offers to manage the full prescription given by the doctors. It would bring the right drug combination for your ailment and help you meet your needs without errors.

• Automated Claims Processing: Of course, the system requires automated claim processing systems, and it keeps an entire record of the claim submissions including adjudication (V5.1, etc)

• Accuracy & Safety: As it is the question of many lives, hence online pharmacy software offers fail-safe features for accuracy & safety, leaving no scope for errors, as it will ensure the best output and there will be kept the effort high to safe the system from any error.

• Complete & Accurate Drug Interaction & Other Databases: The pharmacy database management system is able to detect Drug-Drug Interactions and this way it reduces the chances of ADEs, preserves patients' safety, and prevents related medical and legal problems. A/R Management: Account Receivables Management is crucial in the arena of healthcare to ensure smooth & successful functioning and that the owed amount is refunded back in the shortest time possible. It looks after denied claims and reopens them to receive maximum reimbursement.

• On-line Reconciliation: It's a cost-effective feature to increase cash flow, reduce write-offs and costs. This is a vital feature of the pharmacy management program in increasing profits & revenues.

• Drug-file Updates: All the updates regarding the new drugs introduced in the market and other drugs related information can be found in the system, these file will be kept by the system in backup as there will be always a backup use, so pharmacy can use this data future to make decision and to take fruitful actions for client and their own profit.

• Auto Dr. Fax: It allows your doctor to call in or fax the patient's refill to the pharmacist, saving the patient's trip to the Pharmacy. • Tailored Labels: This feature facilitates customized prescription drug labels, hence making the handlings a lot easier.

• Electronic Wholesaler Ordering: This feature ensures up-to-date costing and inventory, which ties into reliable margin reporting and inventory valuations.

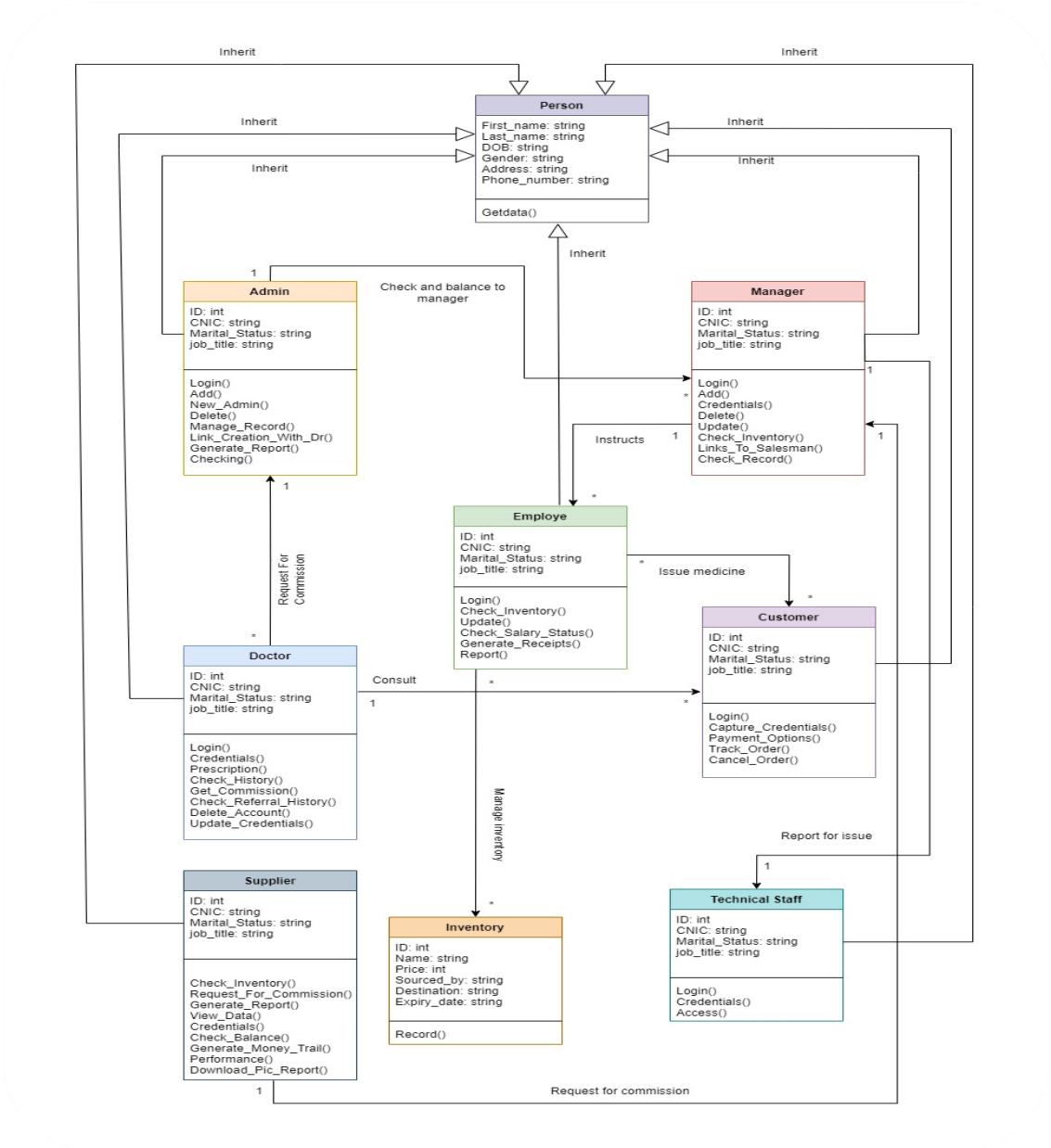
• Reporting: As the month ends, you should be able to tally the revenues generated by the stock movement. You need to be informed of the sale of each unit of an inventory, also what has been disposed of or damaged. Hence you can ensure that you don't lose even a penny without it being accounted for.

2.1.7. Identifying External Entities

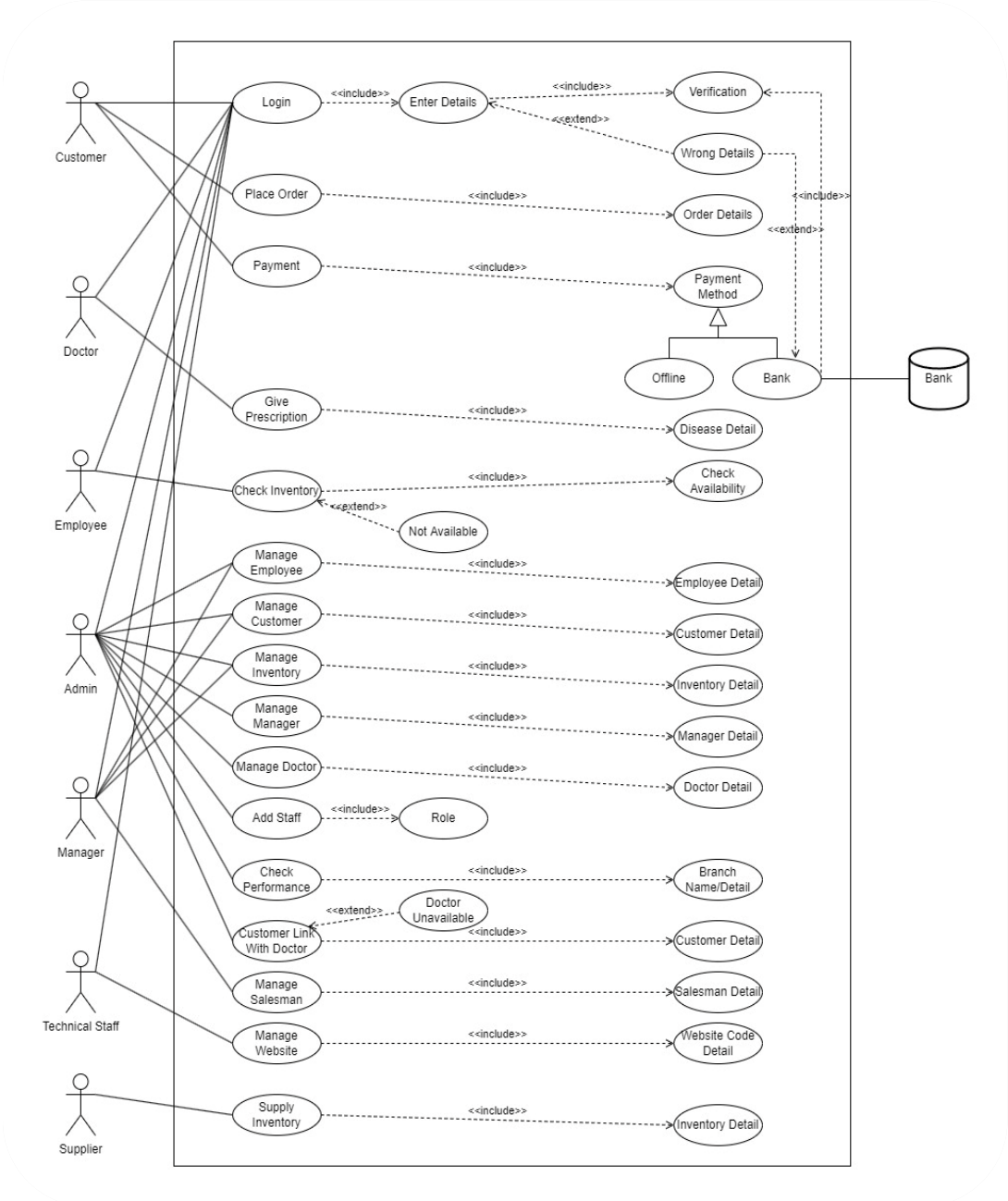
External entities are representations of data provided by services in other applications. These entities enable low-code developers to work with data from other applications in the same way they work with data from their own application – by using Mendix retrieves and XPath expressions, and by directly connecting external entities to page widgets.

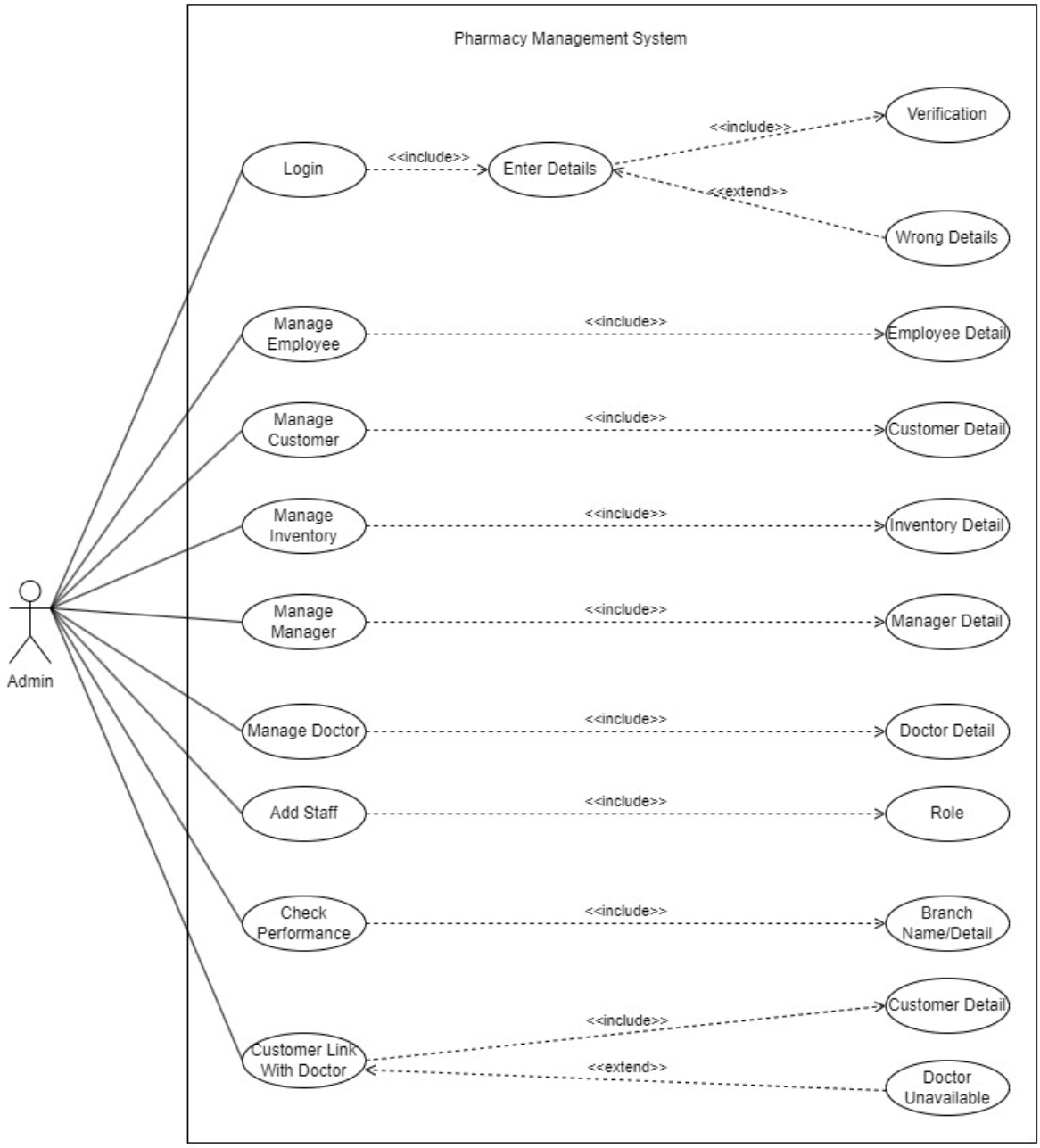
The Identification of External Entities is done in two phases.

2.1.8. Class diagram

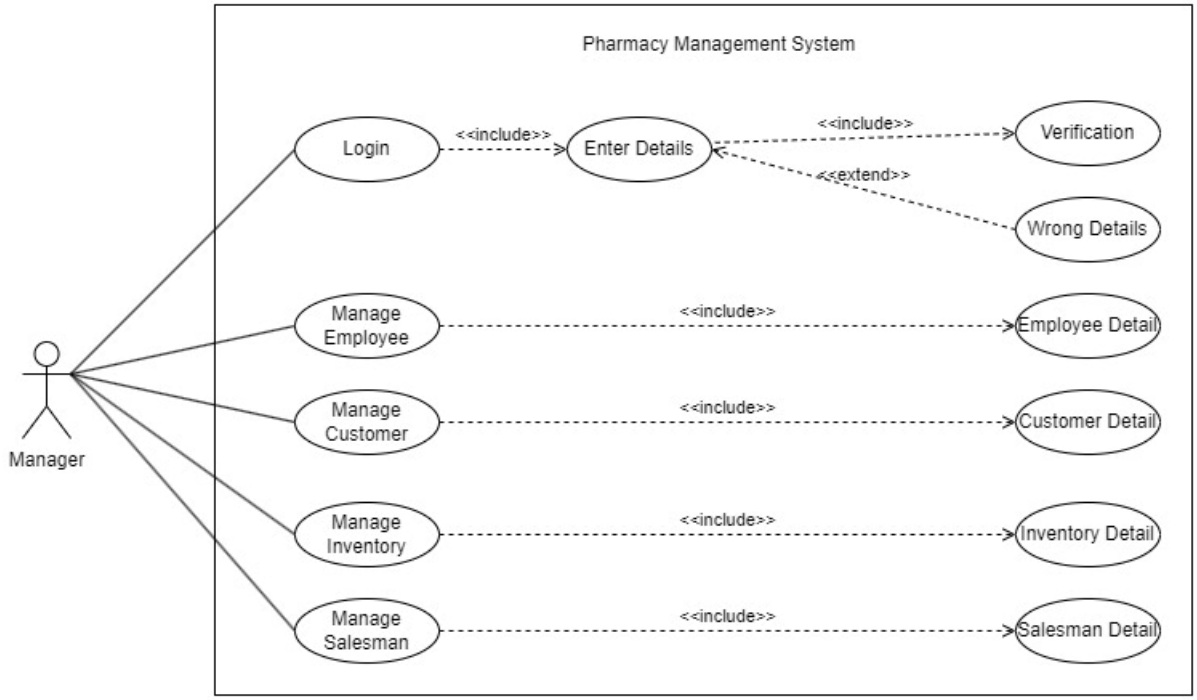


2.1.9. Use case diagram

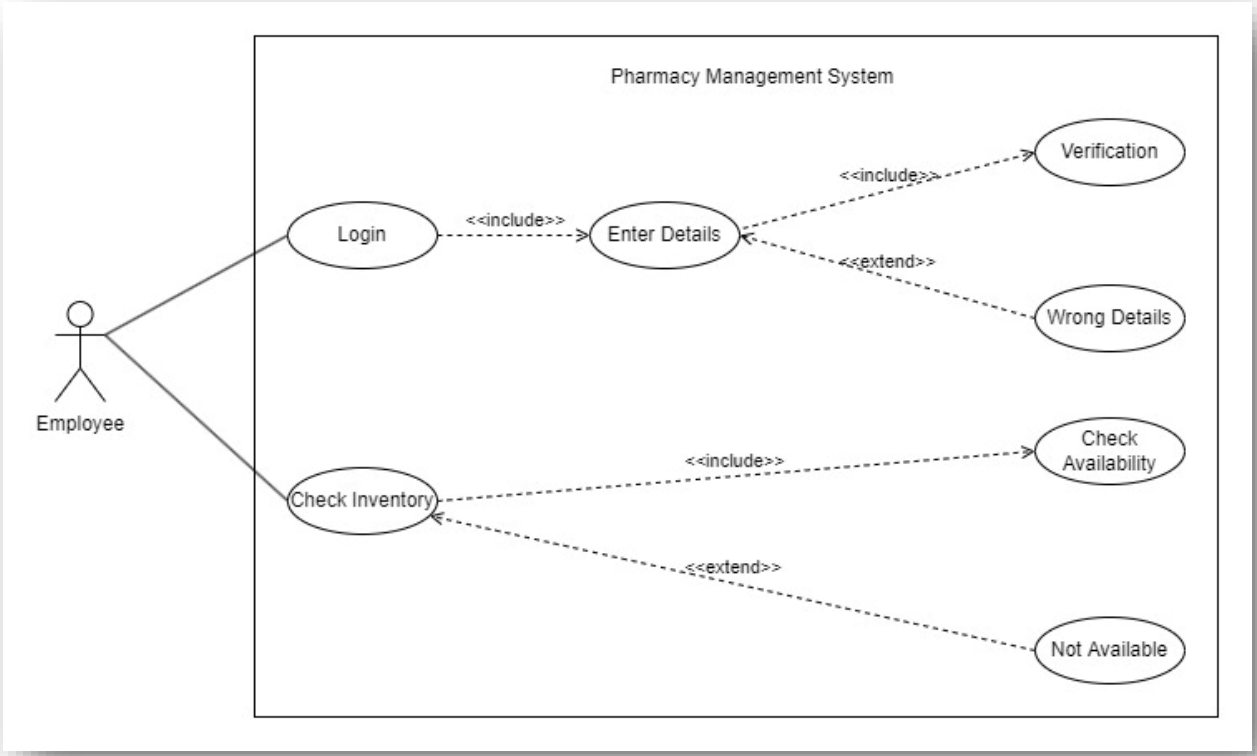




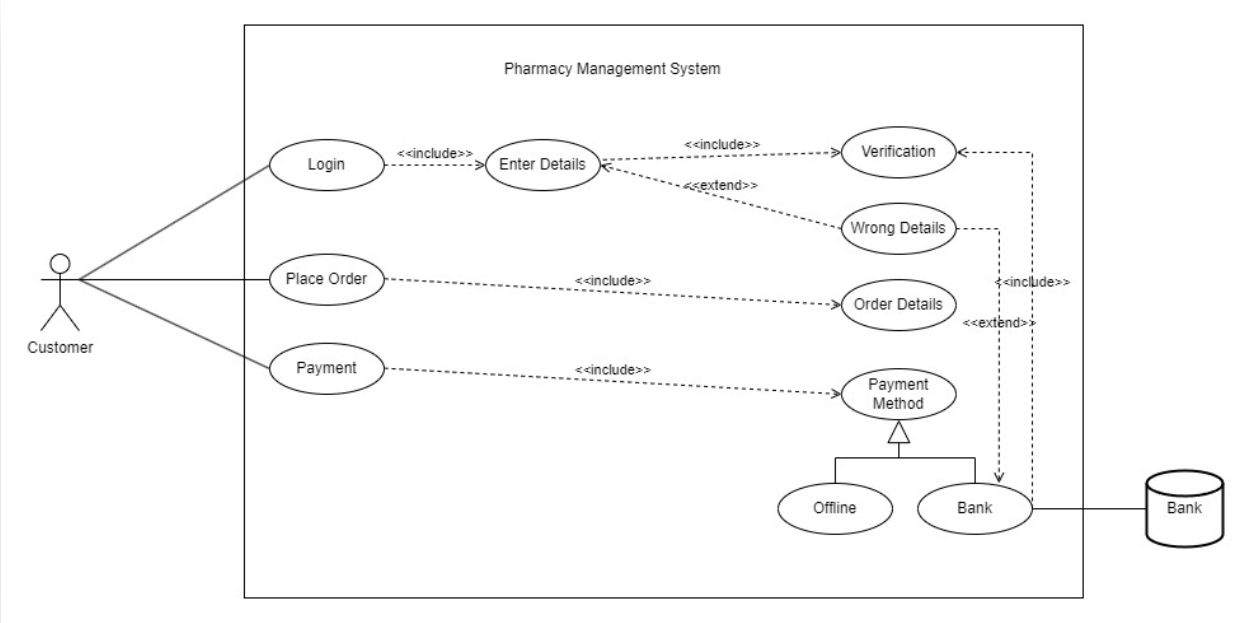
Manager class in use case diagram



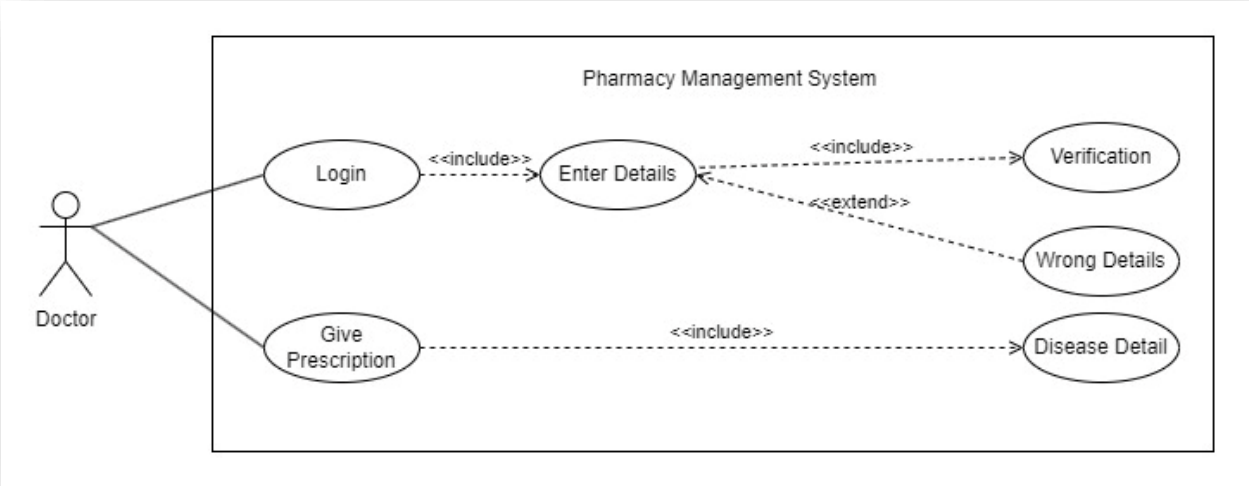
Employee class in use case diagra



Client class in use case diagram

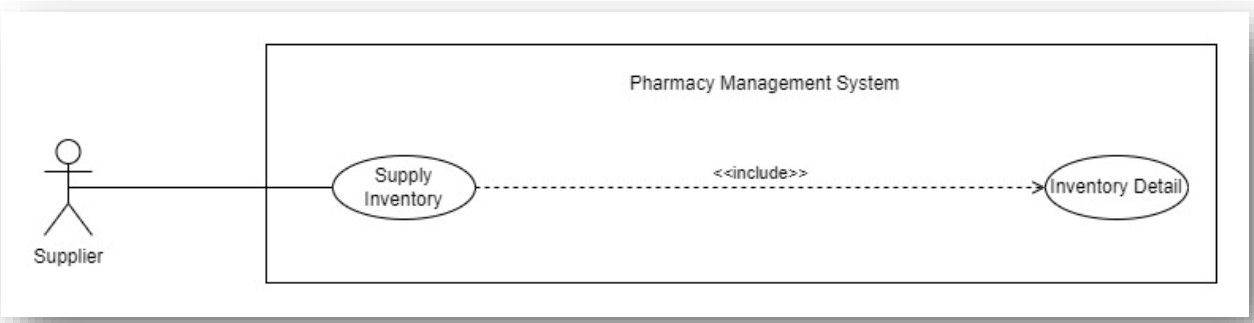


Doctors class in use case diagram

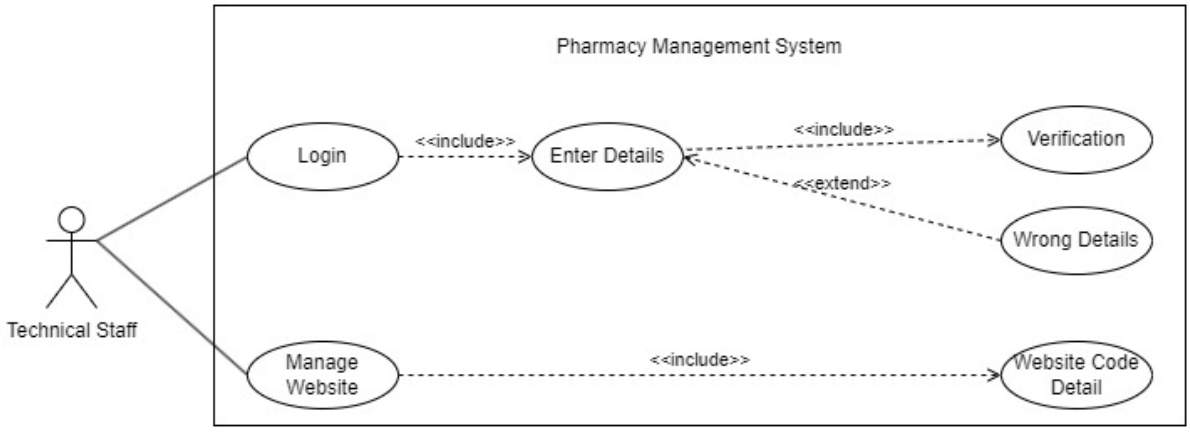


Doctor class in use case diagrams

Supplier class in use case diagram



Technical staff class in use case diagram



3.1. Initial Requirements for all classes

3.1.1. Requirements for the Admin class:

Para	Initial Requirements
1.0	Admin ” Shall ” login to system for the use.
1.0	Admin ” Shall ” use the name and password.
1.0	System “ Shall ” be operated only by one person as admin.
1.0	Admin” Shall” add the people to system.
1.0	New admin ”Shall “ need of previous admin for the login.
1.0	Admin “Shall” add the staff to system.
1.0	Admin” Shall” delete the record of staff.
1.0	Admin “ Shall” the record of staff in system.
1.0	Admin” Shall” make the link to doctors as well on need of time.
1.0	Admin “ Shall’ generate a report for analysis.
1.0	Admin “ Shall” check the performance of any branch of pharmacy.
1.0	Admin “ Shall” generate a report for any period of time.

3.1.2. Initial Requirements for the Manager class:

Para # 02	Initial Requirements
2.0	Manager “Shall” login to system for the use.
2.0	Manager “Shall” must have unique name and password to system for the use.
2.0	Manager “Shall” add the employees.
2.0	Manager “Shall” have credentials like name, password, address, phone number , Unique id etc.
2.0	Manager “Shall” delete the employee from the record.
2.0	Manager “Shall” Update the employee record to system.
2.0	Manager “Shall” check the Inventory.
2.0	Manager “Shall” make the links to salesmen.
2.0	Manager “Shall” check the record of more than one pharmacy.
2.0	System “Shall” capture the every activity of manager within the system using credentials.

3.1.3. Initial Requirements for the Employee Class:

Para # 03	Initial requirements
3.0	Employee “Shall” have to login for the use of system.
3.0	Employee “Shall” have the name and password to login for the use of system.
3.0	Employee “Shall” check the Inventory.
3.0	Employee “Shall” update the record of inventory as according to need.
3.0	Employee “Shall” have name, password, phone number, working address, unique id etc.
3.0	Employee “Shall” check the salary status.
3.0	Employee “Shall” generate the receipts.
3.0	Employee “Shall” not change name and password in system.
3.0	Employee “Shall” send a report to manager.
3.0	Employee “Shall” change any part of inventory, but credentials will be saved.

3.1.4. Initial Requirements for the Clients Class:

Para # 04	Initial Requirements
4.0	Client “Shall” must have the login for the system use.
4.0	Client “Shall” must have the name and password for login.
4.0	System “Shall” capture the credentials of client at the time of buying.
4.0	Client “Shall” must have the unique id in system.
4.0	Client “Shall” can join through email or using google chrome.
4.0	Client “Shall” must have the login for the system use.
4.0	Client “Shall” have the option to pay through credit/debit cards.
4.0	Client “Shall” track his order in the system.
4.0	Client “Shall” must have the option to cancel the order within 24 hours.

3.1.5. Initial Requirements for the Doctor class:

Para # 05	Initial Requirements
5.0	Doctor “Shall” generate the account by approval of admin.
5.0	Doctor “Shall” have to login to system.
5.0	Doctor “Shall” have name and password for login.
5.0	Doctor “Shall” have credentials like name, password, address, phone number, unique id etc.
5.0	Doctor “Shall” prescribe the medicines to clients through system.
5.0	Doctor “Shall” check the history of patients.
5.0	Doctor “Shall” get the commission by pharmacy.
5.0	Doctor “Shall” have the option to download receipts for the commission.
5.0	Doctor “Shall” see the referral history.
5.0	Doctor “Shall” have the option to delete account.
5.0	Doctor “Shall” update the credentials in system with approval of admin.

3.1.6. Initial Requirements for the Salesmen class:

Para # 06	Initial Requirements
6.0	Salesman “Shall” login to system.
6.0	Salesman “Shall” have name and password for use.
6.0	Salesman “Shall” check the inventory.
6.0	Salesman “Shall” request for the commission.
6.0	Salesman “Shall” generate the report for use.
6.0	Salesman “Shall” view the data only.
6.0	Salesman “Shall” have credentials just like as others.
6.0	Salesman “Shall” check balance with linked pharmacy.
6.0	Salesman “Shall” generate money trail from system.
6.0	Salesman “Shall” have the performance number in system(highest to lowest).
6.0	Salesman “Shall” download picture and report of medicine.

3.1.7. Initial Requirements for the Inventory class:

Para # 07	Initial Requirements
7.0	Inventory “Shall” have a unique id.
7.0	System “Shall” record date for the each medicine.
7.0	System “Shall” record name, price, source, and destination of inventory.
7.0	Inventory “Shall” changed by staff.
7.0	Inventory “Shall” have a record in system.
7.0	Inventory “Shall” mark sold out by staff.
7.0	Inventory “Shall” be keep in add to cart/wish list.
7.0	Inventory “Shall” show the discount on price.
7.0	System “Shall” record the sold medicines for future use.
7.0	Inventory “Shall” sorted as sold more and less medicines.
7.0	Inventory “Shall” visible through website.
7.0	Inventory “Shall” have a picture for showcasing.
7.0	Inventory “Shall” be ordered according to requirement of client.
7.0	Inventory “Shall” prompts a message before two (2) weeks of inventory ending.

3.1.8. Initial Requirements for Technical staff:

Para # 08	Initial Requirements
8.0	Technical staff “Shall” have login to enter in system.
8.0	Technical staff “Shall” have complete credentials like name, password, address, phone number, unique id, etc.
8.0	Technical staff “Shall” have the access to code of website.
8.0	Technical staff “Shall” have to maintain the system.

3.2. Allocate Requirements:

Allocate Requirements for the Admin class

Para # 01	Initial Requirements	Use Case Name
1.0	An admin “Shall” login to the system.	UC_login
1.0	System “Shall” check credentials for login.	UC_verification
1.0	Admin “Shall” add the staff to system.	UC_add_staff
1.0	New admin “Shall” need of permission to create login for system as admin.	UC_new_admin
1.0	Admin “Shall” delete the staff.	UC_delete_staff
1.0	Admin “Shall” keep the record of staff in system.	UC_record_staff
1.0	Admin “Shall” generate a report for the use from the system.	UC_generate_report
1.0	Admin “Shall” keep the track of performance of all the pharmacy branches.	UC_performance_tracking
1.0	Admin “Shall” can get any report for the period of time.	UC_report_at_time
1.0	System “Shall” give a report to only the admin.	UC_admin_report

Allocate Requirements for the Manager class

Para # 02	Initial Requirement	Use Case Name
2.0	Manager “Shall” login to system for the use.	UC_login
2.0	System “Shall” give a verification and permission for use.	UC_Verification
2.0	Manager “Shall” add the employees.	UC_add_employee
2.0	Manager “Shall” have credentials like name, password, address, phone number , Unique id etc.	UC_crdentials
2.0	Manager “Shall” delete the employee from the record.	UC_delete_employee
2.0	System “Shall” delete the employee. Manager “Shall” Update the employee record to system.	UC_deletion_confirmation
2.0	Manager “Shall” check the Inventory.	UC_inventory_Check
2.0	Manager “Shall” make the links to salesmen.	UC_link _creation
2.0	Manager “Shall” check the record of more than one pharmacy.	UC_record_check
2.0	System “Shall” capture the every activity of manager within the system using credentials.	UC_activity_capturing
2.0	System “Shall” capture the complete data	UC_data_capturing

Allocate Requirements for the Employee Requirements

Para # 03	Initial Requirements	Use case Names
3.0	Employee “Shall” have to login for the use of system.	UC_login
3.0	System “Shall” verified the account of user.	UC_verification
3.0	Employee “Shall” check the Inventory.	
3.0	Employee “Shall” update the record of inventory as according to need.	UC_updation
3.0	System “Shall” update the record and will send an acknowledgement messge.	UC_acknowledgement
3.0	Employee “Shall” have name, password, phone number, working address, unique id etc.	UC_credentials
3.0	Employee “Shall” check the salary status.	UC_salary_record
3.0	Employee “Shall” generate the receipts.	UC_receipts
	System “Shall” generate a message for the receipts.	UC_acknowledgement
3.0	Employee “Shall” not change name and password in system.	UC_nochange
3.0	Employee “Shall” send a report to manager.	UC_report_sending
3.0	Employee “Shall” change any part of inventory, but credentials will be saved.	UC_inventory_Change

Allocate Requirement for the Client Requirements

Para # 04	Initial Requirements	Use case Names
4.0	Client “Shall” must have the login for the system use.	UC_login
4.0	System “Shall” verified the login to enter in system.	UC_verification
4.0	Client “Shall” must have the name and password for login.	UC_account
4.0	System “Shall” capture the credentials of client at the time of buying.	UC_credentials
4.0	Client “Shall” must have the unique id in system.	UC_id
4.0	Client “Shall” can join through email or using google chrome.	UC_google_login
4.0	System “Shall” lead the user to main page for the use.	UC_page_leadig
4.0	Client “Shall” must have the login for the system use.	UC_usage_login
4.0	Client “Shall” have the option to pay through credit/debit cards.	UC_payment
4.0	Client “Shall” track his order in the system.	UC_track_order
4.0	Client “Shall” must have the option to cancel the order within 24 hours.	Uc-cancel
4.0	System “Shall” prompts the message of cancellation.	UC_cancellation_msg

Allocate Requirements for the Doctor class:

Para # 05	Initial Requirements	Use Case Names
5.0	Doctors “shall” have to login to use the system.	UC_login_doctor
5.0	System ”Shall” check the authentication.	UC_verification
5.0	System “ shall” record the credentials of doctor.	UC_credentials_verification
5.0	Doctor “shall” prescribe the medicine to client.	UC_prescription
5.0	System “Shall” verify the system verify the prescribed medicines.	UC_verified_orders
5.0	Doctor “shall” check the history of patient.	UC_history_checking
5.0	Doctor “shall” ask for the commission.	UC_commission_asking
5.0	System “shall” passes the request to admin for the commission.	UC_request_passing
5.0	doctor “shall” delete the account.	UC_account_deletion
5.0	System “shall” delete the account and will send a acknowledgement message.	UC_acknowledgement_msg
5.0	Doctor “shall” update the credentials.	UC_credentials_upgradation
5.0	System “shall” upgrade the credentials and will notify to admin.	UC_upgradation_acknowledgement

Allocate Requirements for the Salesmen class:

Para # 06	Initial Requirements	Use Case Names
6.0	Salesmen “shall” have to login to use the system.	UC_login_salesmen
6.0	System ”Shall” check the authentication.	UC_verification
6.0	System “ shall” record the credentials of salesmen.	UC_credentials_verification
6.0	Salesmen “shall” check the inventory.	UC_inventory_checking
6.0	Salesmen “Shall” request for the commission.	UC_request_sending
6.0	Doctor “shall” report generate.	UC_history_checking
6.0	Doctor “shall” ask for the commission.	UC_commission_asking
6.0	System “shall” passes the request to admin for the commission.	UC_request_passing
6.0	Salesmen “shall” view the data.	UC_data_view
6.0	System “shall” generate the money trail.	UC_trail_generation
6.0	Salesmen “shall” download the image from website for use.	UC_download_allowed
6.0	System “shall” allow the salesmen to download inventory image.	UC_image_link_generation

Allocate Requirements for the inventory class:

Para # 07	Initial Requirements	Use Case Names
7.0	Inventory “shall” have to login to use the system.	UC_login_inventory
7.0	System ”Shall” check the authentication.	UC_verification
7.0	System “ shall” record the credentials of salesmen.	UC_credentials_verification
7.0	inventory “shall” changed by the employee.	UC_inventory_changing
7.0	Inventory “shall” be marked as sold by staff.	UC_marked_sold
7.0	System “shall” marked the sold out and will send the acknowledgement.	UC_acknowledgement
7.0	System “shall” shows the discounted medicines.	UC_discounted_medicines
7.0	Inventory “shall” have the image to show on system.	UC_request_image

Allocate Requirements for the class:

Para # 08	Initial Requirements	Use Case Names
8.0	Technical staff “shall” have to login to use the system.	UC_login_tech_staff
8.0	System ”Shall” check the authentication.	UC_verification
8.0	Tech staff “shall” work according to orders.	UC_orders
8.0	Tech staff “shall” change the code	UC_changing

Red presents the highest priority:

All the requirement which were colored in red shade will be executed on the top of priority and team will complete this work first of all and deliver it as soon as early as possible.

Yellow presents the medium priority.

All the requirement which were colored in yellow shade will be executed on the mid range of priority and team will complete this work after completing the first phase and the delivery of highest priority and then these will requirements will be delivered.

Green presents the lowest priority.

All the requirement which were colored in Green shade will be executed on the last range of priority and team will complete this work after completing the second phase and the delivery of highest, medium priority and then these will requirements will be delivered, team can develop these requirements at the end as these are not required at the early stages of work.

3.3. Prioritize the Requirements

Admin class

Para # 01	Rank	Initial Requirements	Use_case_ID	Use Case Name
1.0	Highest	An admin “Shall” login to the system.	UC_1	UC_login
1.0	Highest	System “Shall” check credentials for login.	UC_2	UC_verification
1.0	Highest	Admin “Shall” add the staff to system.	UC_3	UC_add_staff
1.0	Lowest	New admin “Shall” need of permission to create login for system as admin.	UC_4	UC_new_admin
1.0	Highest	Admin “Shall” delete the staff.	UC_5	UC_delete_staff
1.0	Medium	Admin “Shall” keep the record of staff in system.	UC-6	UC_record_staff
1.0	Medium	Admin “Shall” generate a report for the use from the system.	UC_7	UC_generate_report
1.0	Low	Admin “Shall” keep the track of performance of all the pharmacy branches.	UC_8	UC_performance_tracking
1.0	Low	Admin “Shall” can get any report for the period of time.	UC_9	UC_report_at_time
1.0	medium	System “Shall” give a report to only the admin.	UC_10	UC_admin_report

Manager class		Initial Requirement	Use_case_ID	Use Case Name
2.0	Highest	Manager “Shall” login to system for the use.	UC_1	UC_login
2.0	Highest	System “Shall” give a verification and permission for use.	UC_2	UC_Verification
2.0	Highest	Manager “Shall” add the employees.	UC_3	UC_add_employee
2.0	lowest	Manager “Shall” have credentials like name, password, address, phone number , Unique id etc.	UC_4	UC_crdentials
2.0	Highest	Manager “Shall” delete the employee from the record.	UC_5	UC_delete_employee
2.0	Highest	System “Shall” delete the employee. Manager “Shall” Update the employee record to system.	UC_6	UC_deletion_confirmation
2.0	Medium	Manager “Shall” check the Inventory.	UC_7	UC_inventory_Check
2.0	Medium	Manager “Shall” make the links to salesmen.	UC_8	UC_link _creation
2.0	Low	Manager “Shall” check the record of more than one pharmacy.	UC_9	UC_record_check
2.0	Low	System “Shall” capture the every activity of manager within the system using credentials.	UC_10	UC_activity_capturing
2.0	low	System “Shall” capture the complete data	UC_11	UC_data_capturing

Employee class

Para # 03	Rank	Initial Requirements	Use_case_ID	Use case Names
3.0	Highest	Employee “Shall” have to login for the use of system.	UC-1	UC_login
3.0	Highest	System “Shall” verified the account of user	UC_2	UC_verification
3.0	Highest	Employee “Shall” check the Inventory.	UC_3	
3.0	Highest	Employee “Shall” update the record of inventory as according to need.	UC_4	UC_updation
3.0	Medium	System “Shall” update the record and will send an acknowledgement messge.	UC_5	UC_acknowledgement
3.0	Medium	Employee “Shall” have name, password, phone number, working address, unique id etc.	UC_6	UC_credentials
3.0	Medium	Employee “Shall” check the salary status.	UC_7	UC_salary_record
3.0	Medium	Employee “Shall” generate the receipts.	UC_8	UC_receipts
	Medium	System “Shall” generate a message for the receipts.	UC_9	UC_acknoRankwledgement
3.0	Low	Employee “Shall” not change name and password in system.	UC_10	UC_nochange
3.0	Low	Employee “Shall” send a report to manager.	UC_11	UC_report_sending
3.0	Low	Employee “Shall” change any part of inventory, but credentials will be saved.	UC_12	UC_inventory_Change

Client class

Para # 04	Rank	Initial Requirements	Use_case_ID	Use case Names
4.0	Highest	Client “Shall” must have the login for the system use.	UC_1	UC_login
4.0	Highest	System “Shall” verified the login to enter in system.	UC_2	UC_verification
4.0	Highest	Client “Shall” must have the name and password for login.	UC_3	UC_account
4.0	medium	System “Shall” capture the credentials of client at the time of buying.	UC_4	UC_credentials
4.0	Medium	Client “Shall” must have the unique id in system.	UC_5	UC_id
4.0	Lowest	Client “Shall” can join through email or using google chrome.	UC_6	UC_google_login
4.0	Lowest	System “Shall” lead the user to main page for the use.	UC_7	UC_page_leadig
4.0	Highest	Client “Shall” must have the login for the system use.	UC_8	UC_usage_login
4.0	Lowest	Client “Shall” have the option to pay through credit/debit cards.	UC_9	UC_payment
4.0	Lowest	Client “Shall” track his order in the system.	UC_10	UC_track_order
4.0	Lowest	Client “Shall” must have the option to cancel the order within 24 hours.	UC_11	Uc-cancel
4.0	lowest	System “Shall” prompts the message of cancellation.	UC_12	UC_cancellation _msg

Doctors class

Para # 05	Rank	Initial Requirements	Use_case_ID	Use Case Names
5.0	Highest	Doctors “shall” have to login to use the system.	UC_1	UC_login_doctor
5.0	Highest	System ”Shall” check the authentication.	UC_2	UC_verification
5.0	Medium	System “ shall” record the credentials of doctor.	UC_3	UC_credentials_verification
5.0	Medium	Doctor “shall” prescribe the medicine to client.	UC_4	UC_prescription
5.0	Medium	System “Shall” verify the system verify the prescribed medicines.	UC_5	UC_verified_orders
5.0	Medium	Doctor “shall” check the history of patient.	UC_6	UC_history_checking
5.0	Lowest	Doctor “shall” ask for the commission.	UC_7	UC_commission_asking
5.0	Lowest	System “shall” passes the request to admin for the commission.	UC_8	UC_request_passing
5.0	Lowest	doctor “shall” delete the account.	UC_9	UC_account_deletion
5.0	Lowest	System “shall” delete the account and will send a acknowledgement message.	UC_10	UC_acknowledgement_msg
5.0	Lowest	Doctor “shall” update the credentials.	UC_11	UC_credentials_upgradation
5.0	lowest	System “shall” upgrade the credentials and will notify to admin.	UC_12	UC_upgradation_acknowledgement

Salesmen class

Para # 06	Rank	Initial Requirements	Use_case_ID	Use Case Names
6.0	Highest	Salesmen “shall” have to login to use the system.	UC_1	UC_login_salesmen
6.0	Highest	System ”Shall” check the authentication.	UC_2	UC_verification
6.0	Lowest	System “ shall” record the credentials of salesmen.	UC_3	UC_credentials_verification
6.0	Highest	Salesmen “shall” check the inventory.	UC_4	UC_inventory_checking
6.0	Lowest	Salesmen “Shall” request for the commission.	UC_5	UC_request_sending
6.0	Lowest	Doctor “shall” report generate.	UC_6	UC_history_checking
6.0	Lowest	Doctor “shall” ask for the commission.	UC_7	UC_commission_asking
6.0	Lowest	System “shall” passes the request to admin for the commission.	UC_8	UC_request_passing
6.0	Lowest	Salesmen “shall” view the data.	UC_9	UC_data_view
6.0	Lowest	System “shall” generate the money trail.	UC_10	UC_trail_generation
6.0	Lowest	Salesmen “shall” download the image from website for use.	UC_11	UC_download_allowed
6.0	lowest	System “shall” allow the salesmen to download inventory image.	UC_12	UC_image_link_generation

Inventory class

Para # 07	Rank	Initial Requirements	Use_case_ID	Use Case Names
7.0	Highest	Inventory “shall” have to login to use the system.	UC_1	UC_login_inventory
7.0	Highest	System ”Shall” check the authentication.	UC_2	UC_verification
7.0	Lowest	System “ shall” record the credentials of salesmen.	UC_3	UC_credentials_verification
7.0	Medium	inventory “shall” changed by the employee.	UC_4	UC_inventory_changing
7.0	Medium	Inventory “shall” be marked as sold by staff.	UC_5	UC_marked_sold
7.0	Medium	System “shall” marked the sold out and will send the acknowledgement.	UC_6	UC_acknowledgement
7.0	Lowest	System “shall” shows the discounted medicines.	UC_7	UC_discounted_medicines
7.0	lowest	Inventory “shall” have the image to show on system.	UC_8	UC_request_image

Technical staff class

Para # 08	Rank	Initial Requirements	Use_case_ID	Use Case Names
8.0	Highest	Inventory “shall” have to login to use the system.	UC_1	UC_login_inventory
8.0	Highest	System ”Shall” check the authentication.	UC_2	UC_verification
8.0	Lowest	System “ shall” record the credentials of salesmen.	UC_3	UC_credentials_verification
8.0	Medium	inventory “shall” changed by the employee.	UC_4	UC_inventory_changing
8.0	Medium	Inventory “shall” be marked as sold by staff.	UC_5	UC_marked_sold
8.0	Medium	System “shall” marked the sold out and will send the acknowledgement.	UC_6	UC_acknowledgement
8.0	Lowest	System “shall” shows the discounted medicines.	UC_7	UC_discounted_medicines
8.0	lowest	Inventory “shall” have the image to show on system.	UC_8	UC_request_image

3.4. Traceability requirement matrix

Sr#	Para # 01	Initial Requirements	Build	Use Case Name	Category
1	1.0	An admin “Shall” login to the system.	B1	UC_login	Business
2	1.0	System “Shall” check credentials for login.	B2	UC_verification	Business
3	1.0	Admin “Shall” add the staff to system.	B3	UC_add_staff	Business
4	1.0	New admin “Shall” need of permission to create login for system as admin.	B4	UC_new_admin	Business
5	1.0	Admin “Shall” delete the staff.	B5	UC_delete_staff	Business
6	1.0	Admin “Shall” keep the record of staff in system.	B6	UC_record_staff	Business
7	1.0	Admin “Shall” generate a report for the use from the system.	B7	UC_generate_report	Business
8	1.0	Admin “Shall” keep the track of performance of all the pharmacy branches.	B8	UC_performance_tracking	Business
9	1.0	Admin “Shall” can get any report for the period of time.	B9	UC_report_at_time	Business
10	1.0	System “Shall” give a report to only the admin.	B10	UC_admin_report	Business

Traceability requirement matrix
Manager class

Sr#		Initial Requirement	Build	Use Case Name	Category
1		Manager “Shall” login to system for the use.	B2	UC_login	Business
2		System “Shall” give a verification and permission for use.	B2	UC_Verification	Business
3		Manager “Shall” add the employees.	B2	UC_add_employee	Business
4		Manager “Shall” have credentials like name, password, address, phone number , Unique id etc.	B2	UC_crdentials	Business
5		Manager “Shall” delete the employee from the record.	B2	UC_delete_employee	Business
6		System “Shall” delete the employee. Manager “Shall” Update the employee record to system.	B2	UC_deletion_confirmation	Business
7		Manager “Shall” check the Inventory.	B2	UC_inventory_Check	Business
8		Manager “Shall” make the links to salesmen.	B2	UC_link _creation	Business
9		Manager “Shall” check the record of more than one pharmacy.	B2	UC_record_check	Business
10		System “Shall” capture the every activity of manager within the system using credentials.	B2	UC_activity_capturing	Business
11		System “Shall” capture the complete data	B2	UC_data_capturing	Business

Traceability requirement matrix
Employee class

Sr#	Para # 03	Initial Requirements	Build	Use case Names	Category
1	3.0	Employee “Shall” have to login for the use of system.	B3	UC_login	Business
2	3.0	System “Shall” verified the account of user	B3	UC_verification	Business
3	3.0	Employee “Shall” check the Inventory.	B3		Business
4	3.0	Employee “Shall” update the record of inventory as according to need.	B3	UC_updation	Business
5	3.0	System “Shall” update the record and will send an acknowledgement messge.	B3	UC_acknowledgement	Business
6	3.0	Employee “Shall” have name, password, phone number, working address, unique id etc.	B3	UC_credentials	Business
7	3.0	Employee “Shall” check the salary status.	B3	UC_salary_record	Business
8	3.0	Employee “Shall” generate the receipts.	B3	UC_receipts	Business
9		System “Shall” generate a message for the receipts.	B3	UC_acknowledgement	Business
10	3.0	Employee “Shall” not change name and password in system.	B3	UC_nochange	Business
11	3.0	Employee “Shall” send a report to manager.	B3	UC_report_sending	Business
12	3.0	Employee “Shall” change any part of inventory, but credentials will be saved.	B3	UC_inventory_Change	Business

Sr#	Para # 04	Initial Requirements	Build	Use case Names	Category

Traceability requirement matrix
Clients class

1	4.0	Client “Shall” must have the login for the system use.	B4	UC_login	Business
2	4.0	System “Shall” verified the login to enter in system.	B4	UC_verification	Business
3	4.0	Client “Shall” must have the name and password for login.	B4	UC_account	Business
4	4.0	System “Shall” capture the credentials of client at the time of buying.	B4	UC_credentials	Business
5	4.0	Client “Shall” must have the unique id in system.	B4	UC_id	Business
6	4.0	Client “Shall” can join through email or using google chrome.	B4	UC_google_login	Business
7	4.0	System “Shall” lead the user to main page for the use.	B4	UC_page_leadig	Business
8	4.0	Client “Shall” must have the login for the system use.	B4	UC_usage_login	Business
9	4.0	Client “Shall” have the option to pay through credit/debit cards.	B4	UC_payment	Business
10	4.0	Client “Shall” track his order in the system.	B4	UC_track_order	Business
11	4.0	Client “Shall” must have the option to cancel the order within 24 hours.	B4	Uc-cancel	Business
12	4.0	System “Shall” prompts the message of cancellation.	B4	UC_cancellation_msg	Business

Traceability requirement matrix
Doctors class

Sr#	Para # 05	Initial Requirements	Build	Use Case Names	Category
1	5.0	Doctors “shall” have to login to use the system.	B5	UC_login_doctor	Business
2	5.0	System ”Shall” check the authentication.	B5	UC_verification	Business
3	5.0	System “ shall” record the credentials of doctor.	B5	UC_credentials_verification	Business
4	5.0	Doctor “shall” prescribe the medicine to client.	B5	UC_prescription	Business
5	5.0	System “Shall” verify the system verify the prescribed medicines.	B5	UC_verified_orders	Business
6	5.0	Doctor “shall” check the history of patient.	B5	UC_history_checking	Business
7	5.0	Doctor “shall” ask for the commission.	B5	UC_commission_asking	Business
8	5.0	System “shall” passes the request to admin for the commission.	B5	UC_request_passing	Business
9	5.0	doctor “shall” delete the account.	B5	UC_account_deletion	Business
10	5.0	System “shall” delete the account and will send a acknowledgement message.	B5	UC_acknowledgement_msg	Business
11	5.0	Doctor “shall” update the credentials.	B5	UC_credentials_upgradation	Business
12	5.0	System “shall” upgrade the credentials and will notify to admin.	B5	UC_upgradation_acknowledgement	Business

Traceability requirement matrix Salesmen class

Sr#	Para # 06	Initial Requirements	Build	Use Case Names	Category
1	6.0	Salesmen “shall” have to login to use the system.	B6	UC_login_salesmen	Business
2	6.0	System ”Shall” check the authentication.	B6	UC_verification	Business
3	6.0	System “ shall” record the credentials of salesmen.	B6	UC_credentials_verification	Business
4	6.0	Salesmen “shall” check the inventory.	B6	UC_inventory_checking	Business
5	6.0	Salesmen “Shall” request for the commission.	B6	UC_request_sending	Business
6	6.0	Doctor “shall” report generate.	B6	UC_history_checking	Business
7	6.0	Doctor “shall” ask for the commission.	B6	UC_commission_asking	Business
8	6.0	System “shall” passes the request to admin for the commission.	B6	UC_request_passing	Business
9	6.0	Salesmen “shall” view the data.	B6	UC_data_view	Business
10	6.0	System “shall” generate the money trail.	B6	UC_trail_generation	Business
11	6.0	Salesmen “shall” download the image from website for use.	B6	UC_download_allowed	Business

12	6.0	System “shall” allow the salesmen to download inventory image.	B6	UC_image_link_generation	Business

Traceability requirement matrix Inventory class

Sr#	Para # 07	Initial Requirements	Build	Use Case Names	Category
1	7.0	Inventory “shall” have to login to use the system.	B7	UC_login_inventory	Business
2	7.0	System ”Shall” check the authentication.	B7	UC_verification	Business
3	7.0	System “ shall” record the credentials of salesmen.	B7	UC_credentials_verification	Business
4	7.0	inventory “shall” changed by the employee.	B7	UC_inventory_changing	Business
5	7.0	Inventory “shall” be marked as sold by staff.	B7	UC_marked_sold	Business
6	7.0	System “shall” marked the sold out and will send the acknowledgement.	B7	UC_acknowledgement	Business

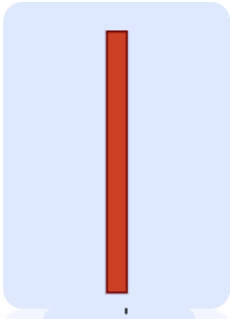
7	7.0	System “shall” shows the discounted medicines.	B7	UC_discounted_medicines	Business
8	7.0	Inventory “shall” have the image to show on system.	B7	UC_request_image	Business

Traceability requirement matrix
Technical staff class

Sr#	Para # 08	Initial Requirements	Build	Use Case Names	Category
1	8.0	Inventory “shall” have to login to use the system.	B8	UC_login_inventory	Business
2	8.0	System ”Shall” check the authentication.	B8	UC_verification	Business
3	8.0	System “ shall” record the credentials of salesmen.	B8	UC_credentials_verification	Business
4	8.0	inventory “shall” changed by the employee.	B8	UC_inventory_changing	Business
5	8.0	Inventory “shall” be marked as sold by staff.	B8	UC_marked_sold	Business
6	8.0	System “shall” marked the sold out and will send the acknowledgement.	B8	UC_acknowledgement	Business

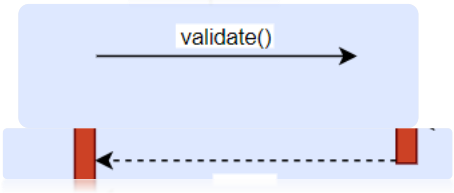
7	8.0	System “shall” shows the discounted medicines.	B8	UC_discounted_medicines	Business
8	8.0	Inventory “shall” have the image to show on system.	B8	UC_request_image	Business

Activation line:
This Is a activation line whenever it starts in sequence diagram it shows that here the working of object has been started.



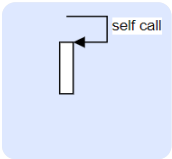
Lifeline:
This shows the life of any object, represents that for how much time an object done the job.

Message passing arrow:
This is a sign we use to pass the message between the object to communicate to each others.

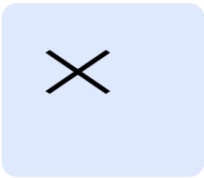


Acknowledgement message:
This arrow shows the acknowledgement by the object.

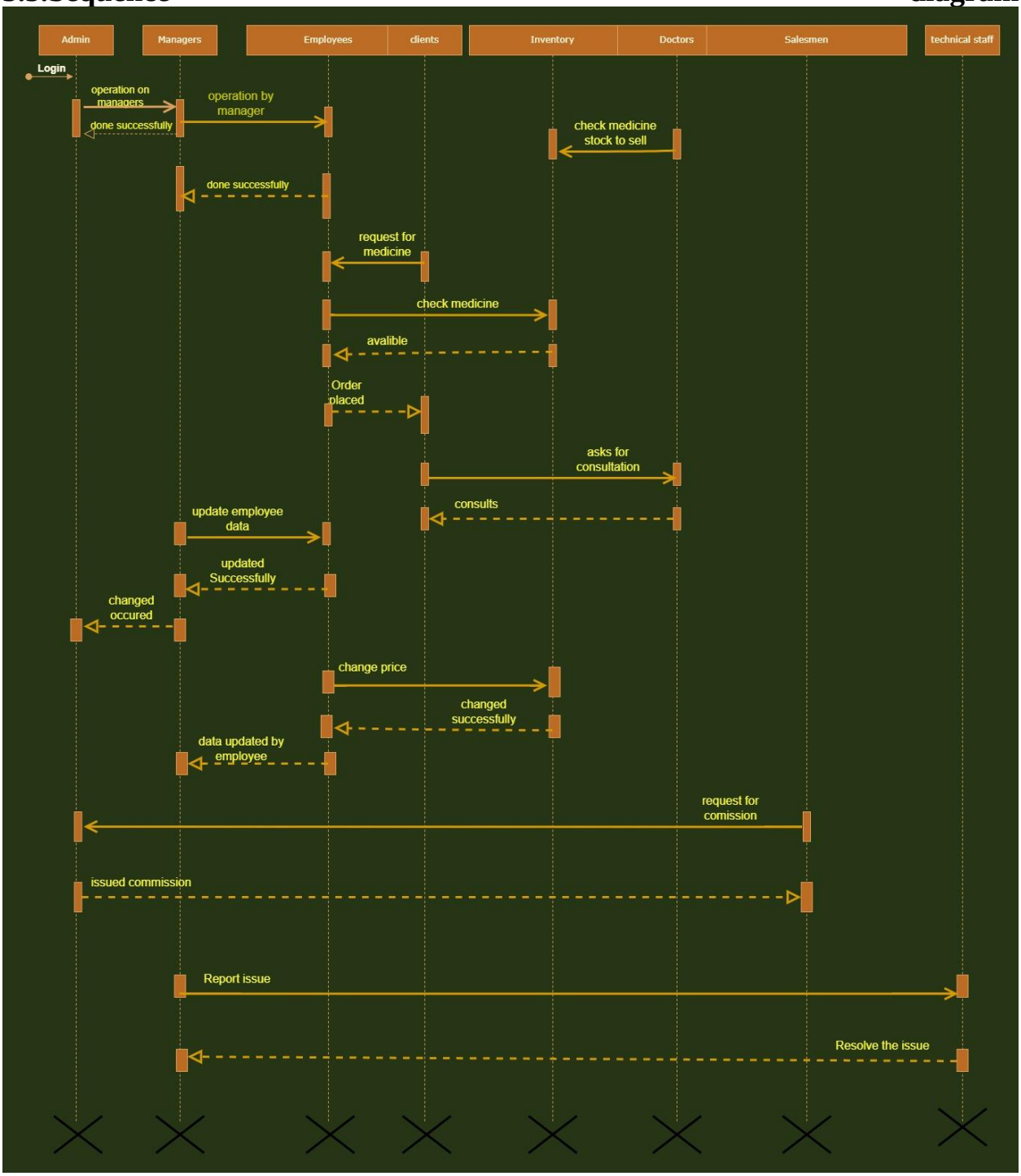
Self call:
This is a self call message, which validate itself before proceeding to next step.



End message:
This sign shows the end of lifeline.



3.5.Sequence diagram



4.1. Introduction:

Third deliverable is all about the software design. In the previous deliverable, analysis of the system is completed. So, we understand the current situation of the problem domain. Now we are ready to strive for a solution for the problem domain by using object-oriented approach. Following artifacts must be included in the 3rd deliverable.

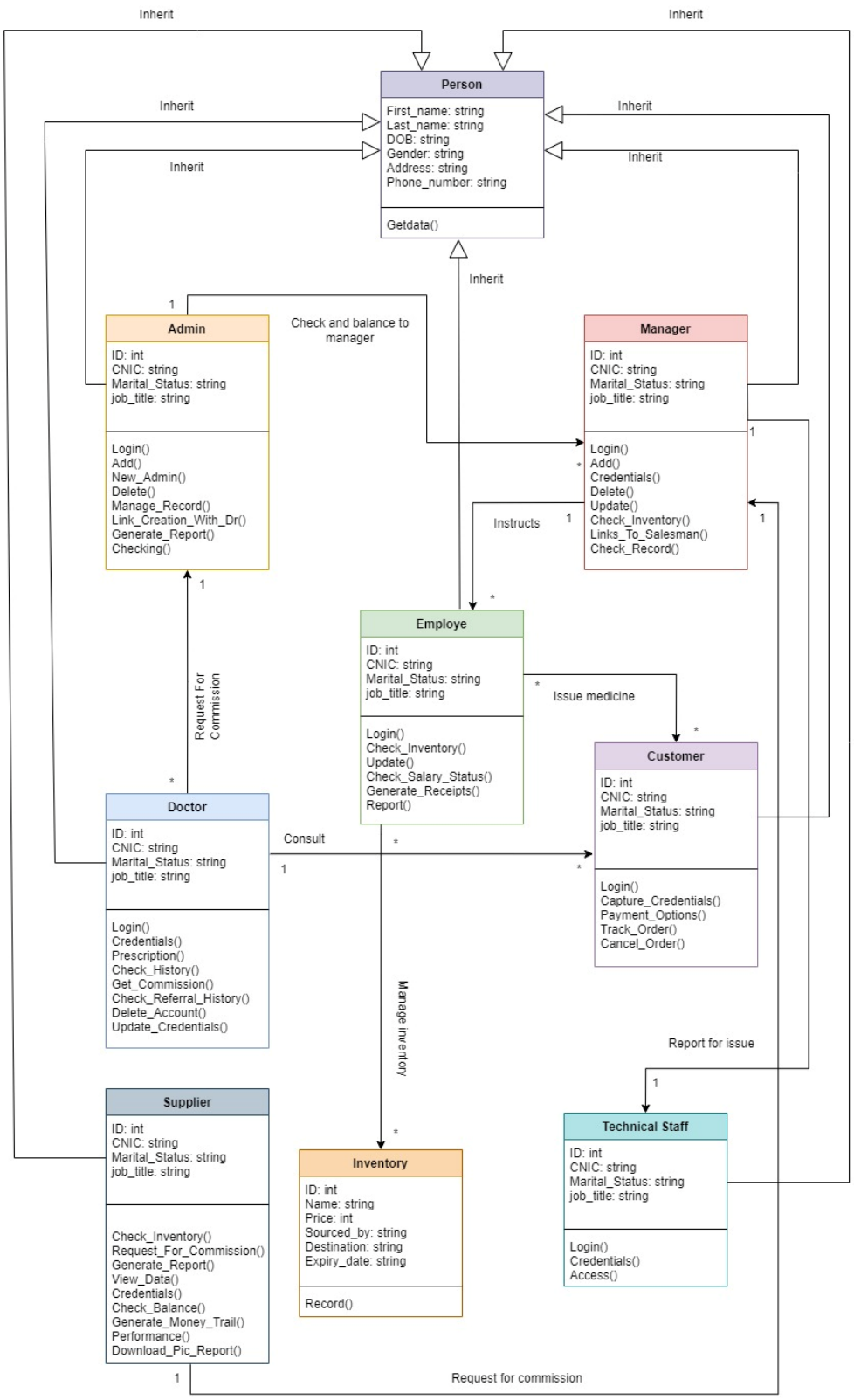
1. Domain Model
2. System Sequence Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. Operation Contracts
6. Design Class Diagram
7. State Transition Diagram
8. Data Model

Now we discuss these artifacts one by one as follows:

4.1.2. Domain Model

Domain models represent the set of requirements that are common to systems within a product line. There may be many domains, or areas of expertise, represented in a single product line and a single domain may span multiple product lines. The requirements represented in a domain model include:

- Definition of scope for the domain
- Information or objects
- Features or use cases, including factors that lead to variation
- Operational/behavioral characteristics


Person
First_name: string
Last_name: string
DOB: string
Gender: string
Address: string
Phone_number: string
Getdata()
Inherit
Admin
ID: int
CNIC: string
Marital_Status: string
job_title: string
Login()
Add()
New_Admin()
Delete()
Manage_Record()
Link_Creation_With_Dr()
Generate_Report()
Checking()
Check and balance to manager
Manager
ID: int
CNIC: string
Marital_Status: string
job_title: string
Login()
Add()
Credentials()
Delete()
Update()
Check_Inventory()
Links_To_Salesman()
Check_Record()
Instructs
Employe
ID: int
CNIC: string
Marital_Status: string
job_title: string
Login()
Check_Inventory()
Update()
Check_Salary_Status()
Generate_Receipts()
Report()
Issue medicine
Request For Commission
Doctor
ID: int
CNIC: string
Marital_Status: string
job_title: string
Login()
Credentials()
Prescription()
Check_History()
Get_Commission()
Check_Referral_History()
Delete_Account()
Update_Credentials()
Consult
Customer
ID: int
CNIC: string
Marital_Status: string
job_title: string
Login()
Capture_Credentials()
Payment_Options()
Track_Order()
Cancel_Order()
Manage inventory
Report for issue
Supplier
ID: int
CNIC: string
Marital_Status: string
job_title: string
Check_Inventory()
Request_For_Commission()
Generate_Report()
View_Data()
Credentials()
Check_Balance()
Generate_Money_Trail()
Performance()
Download_Pic_Report()
Inventory
ID: int
Name: string
Price: int
Sourced_by: string
Destination: string
Expiry_date: string
Record()
Technical Staff
ID: int
CNIC: string
Marital_Status: string
job_title: string
Login()
Credentials()
Access()
Request for commission

4.1.3. What is domain modeling?

According to Rational Unified Process, ® or RUP, ® a domain model is a business object model that focuses on "product, deliverables, or events that are important to the business domain." A domain model is an "incomplete" business model, in that it omits individual worker responsibilities. The point of domain modeling is to provide "the big picture" of the interrelationships among business entities in a complex organization.

It also provides a high-level description of the data that each entity provides. Domain modeling plays a central role in understanding the current environment and planning for the future.

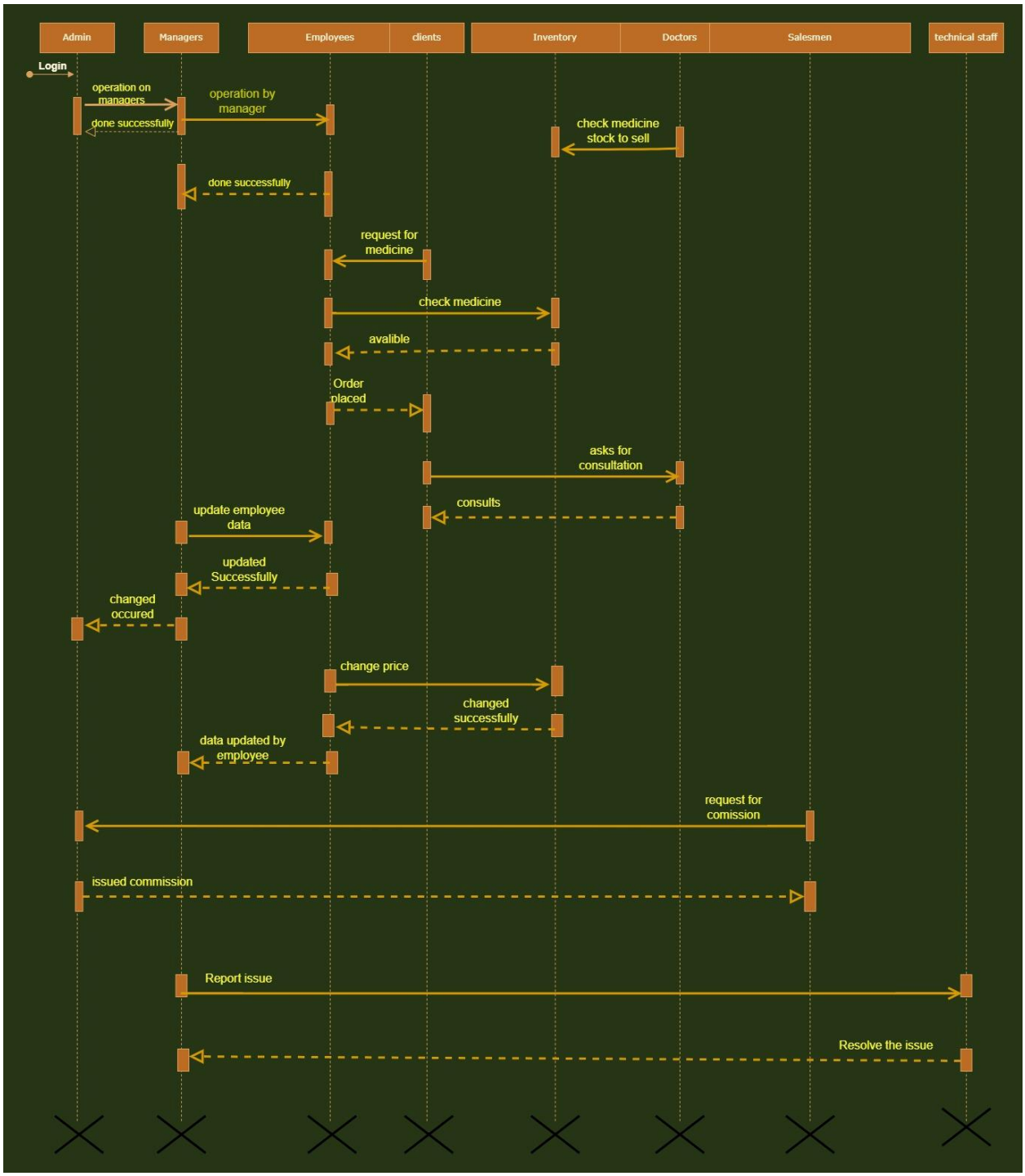
- The typical steps involved in domain modeling are:
- Illustrate meaningful conceptual classes in a real-world problem domain
- Identify conceptual classes or domain objects
- Show associations between them
- Indicate their attributes when appropriate
- Purposely incomplete

4.1.4. System Sequence Diagram

The UML system sequence diagram (SSD) illustrates events sequentially input from an external source to the system. A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behavior of a system.

4.1.5. Sequence Diagram

A sequence diagram shows, a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.


Admin
Managers
Employees
clients
Inventory
Doctors
Salesmen
technical staff
Login
operation on managers
done successfully
operation by manager
check medicine stock to sell
done successfully
request for medicine
check medicine
avalible
Order placed
asks for consultation
consults
update employee data
updated Successfully
changed occured
change price
changed successfully
data updated by employee
request for comission
issued commission
Report issue
Resolve the issue

4.1.6. Collaboration Diagram

A collaboration diagram describes a pattern of interaction among objects; it shows the objects participating in the interaction by their links to each other and the messages that they send to each other.

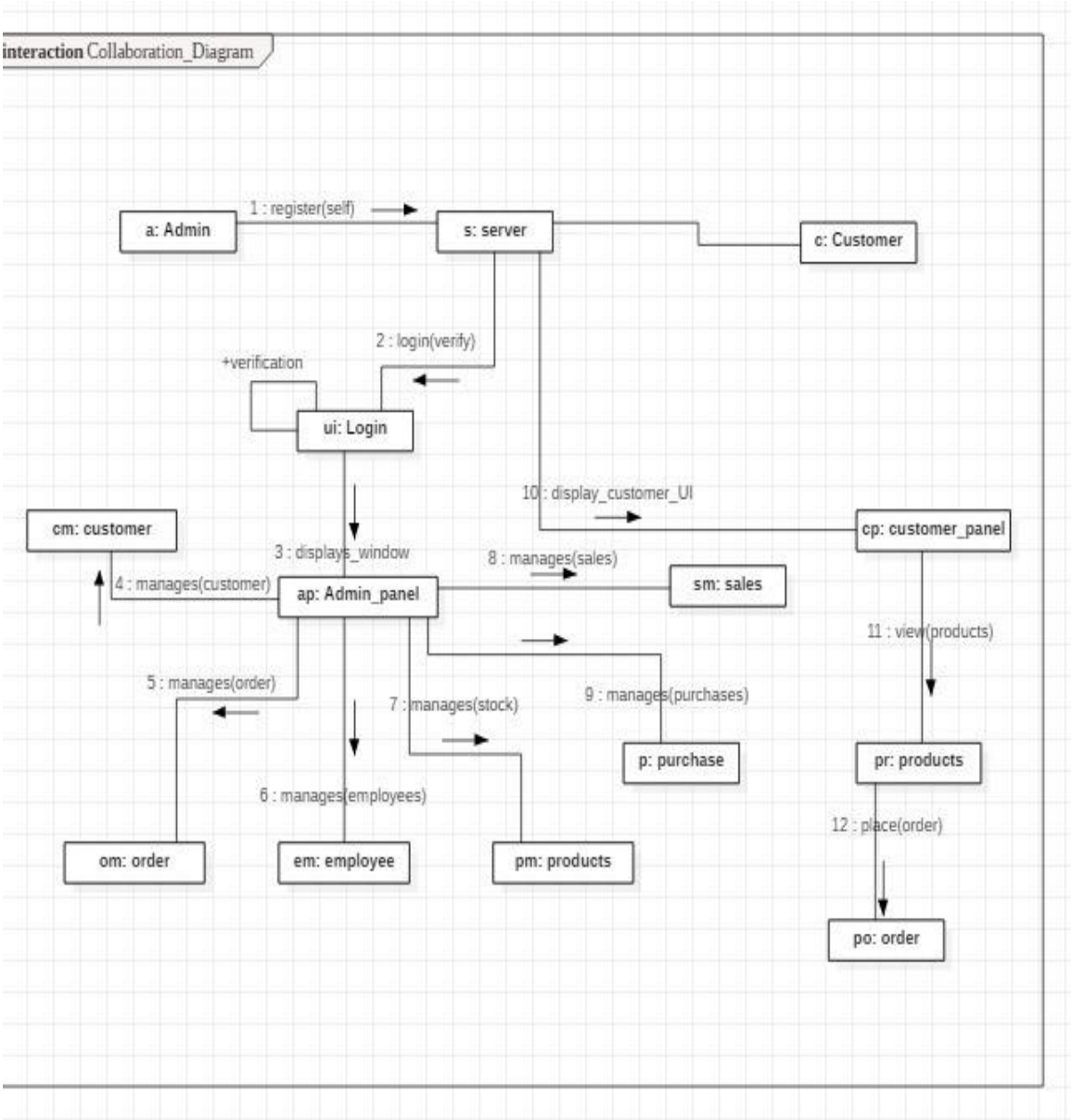


Figure Collaboration Diagram

4.1.7. Operation Contracts

A UML Operation contract identifies system state changes when an operation happens. Effectively, it will define what each system operation does. An operation is taken from a system sequence diagram. It is a single event from that diagram. A domain model can be used to help generate an operation contract.

Table 3.6.1: Register

Name	**Register ()**
Responsibilities	Register an Admin to the system
Cross References	UC_Registration
Pre-Conditions	There is an Admin who is willing to register himself to the System.
Post Conditions	▪ Admin was associated with a database ▪ Admin was assigned with a username and a password. ▪ Admin was registered successfully.
Exceptions	Admin's details are invalid.

Table 3.6.2: Login

Name	**Login ()**
Responsibilities	Admin logins to the system
Cross References	UC_Login
Pre-Conditions	Admin must be registered and an authorized person.
Post Conditions	▪ Username and password were entered by the admin. ▪ Entered credentials were matched with the database. ▪ Admin was logged in successfully.
Exceptions	Admin entered wrong credentials

Table 3.6.3: Add Item

Name	**Add item ()**
Responsibilities	Admin adds an item to the menu of the system.
Cross References	UC_Item Menu
Pre-Conditions	Admin is logged in

Post Conditions	▪ An add item instance was created. ▪ Item was assigned a unique ID. ▪ Items was associated with a database. Items was added successfully.
Exceptions	None

Table 3.6.4: View Item

Name	**View item ()**
Responsibilities	Admin view item menu to the system
Cross References	UC_Item Menu
Pre-Conditions	Admin is logged in
Post Conditions	▪ A view item instance is created. ▪ Admin viewed item menu from database.
Exceptions	None

Table 3.6.5: Delete Item

Name	**Delete Item ()**
Responsibilities	Admin delete an item from item menu to the system.
Cross References	UC_Item Menu
Pre-Conditions	Admin is logged in
Post Conditions	▪ A delete item instance was created Item was deleted by unique ID. ▪ Item was deleted with a database. ▪ Item was deleted successfully.
Exceptions	None

Table 3.6.6: Update Item

Name	**Update Item ()**
Responsibilities	Admin update an item to the system
Cross References	UC_Item Menu

Pre-Conditions	Admin is logged in
Post Conditions	▪ An update item instance was created Item was edit by unique ID. ▪ Item was edit with a database. ▪ Item was edit successfully.
Exceptions	None

Table 3.6.7: View Record

Name	**View Record ()**
Responsibilities	Admin view records of an employee, supplier, customer and stock availability to the system
Cross References	UC_Record Management
Pre-Conditions	Admin is logged in
Post Conditions	▪ A view record instance was created. ▪ All record was assigned a unique ID. ▪ All records was associated with a database. ▪ Records was viewed successfully.
Exceptions	None

Table 3.6.8: Update Record

Name	**Update Record ()**
Responsibilities	Admin update record of an employee, supplier, customer and stock availability to the system
Cross References	UC_Record Management
Pre-Conditions	Admin is logged in
Post Conditions	▪ An update record instance was created. ▪ Record of employee, customer, supplier and stock availability was update by unique ID. ▪ Record of employee, customer, supplier and stock availability was update with a database. ▪ Record of employee, customer, supplier and stock availability was update successfully.
Exceptions	None

Table 3.6.9: Report

Name	**Report ()**
Responsibilities	Admin generate a report.
Cross References	UC_Report Management
Pre-Conditions	Admin is logged in.
Post Conditions	An admin views the report of the sales, purchases, stock and expense
Exceptions	None

Table 3.6.10: Add Inventory

Name	**Add Inventory ()**
Responsibilities	Admin adds inventory to the system
Cross References	UC_Inventory Management
Pre-Conditions	Admin is logged in.
Post Conditions	▪ An add inventory instance was created Each Inventory was assigned a unique ID. ▪ Inventory was associated with a database. ▪ Inventory was added successfully.
Exceptions	None

Table 3.6.11: Update Record

Name	**Update Inventory ()**
Responsibilities	Admin update record of an employee, supplier, customer and stock availability to the system
Cross References	UC_Inventory Management
Pre-Conditions	Admin is logged in.
Post Conditions	▪ An update inventory instance was created. ▪ Each Inventory was update by unique ID. ▪ Inventory was update with a database. ▪ Inventory was update successfully.
Exceptions	None

Table 3.6.12: Confirm Order

Name	**Confirm Order ()**
Responsibilities	Customer confirms his/her order to the system.
Cross References	UC_Order Management
Preconditions	Customer order an item.
Post Conditions	▪ A confirm order instance is created. ▪ An order is confirmed to the database. ▪ Customer successfully confirmed his/her order.
Exceptions	None

Table 3.6.13: Business Report

Name	**Business Report ()**
Responsibilities	Admin generate a business report to the system
Cross References	UC_Manage Accounts
Pre-Conditions	Admin is logged in.
Post Conditions	A business report is generated.
Exceptions	None

4.1.8. Design Class Diagram

Classes are the work-horses of the design effort they actually perform the real work of the system. The other design elements subsystems, packages and collaborations simply describe how classes are grouped or how they interoperate.

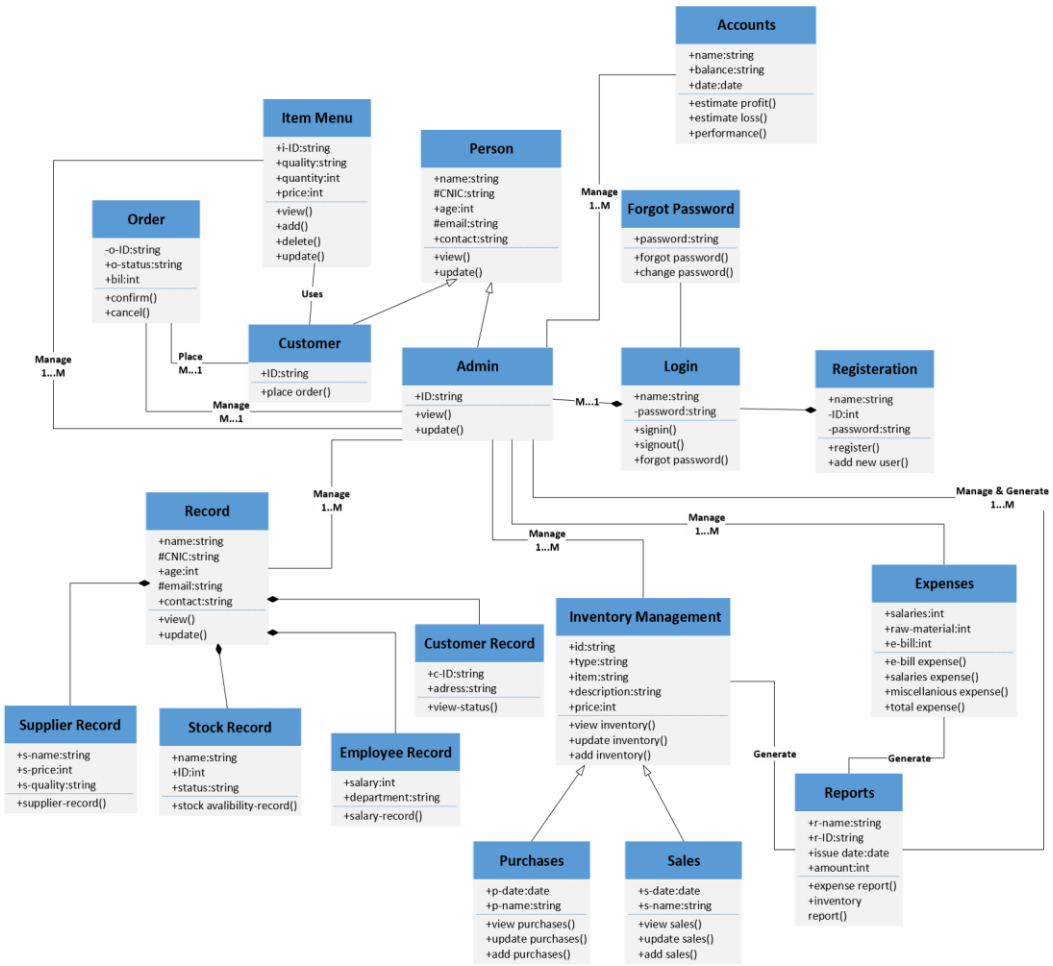


Figure 3.7: Class Diagram

4.1.9. State Chart Diagram

For some operations, the behavior of the operation depends upon the state the receiver object is in. A state machine is a tool for describing the states the object can assume and the events that cause the object to move from one state to another. State machines are most useful for describing active classes. The use of state machines is particularly important for defining the behavior.

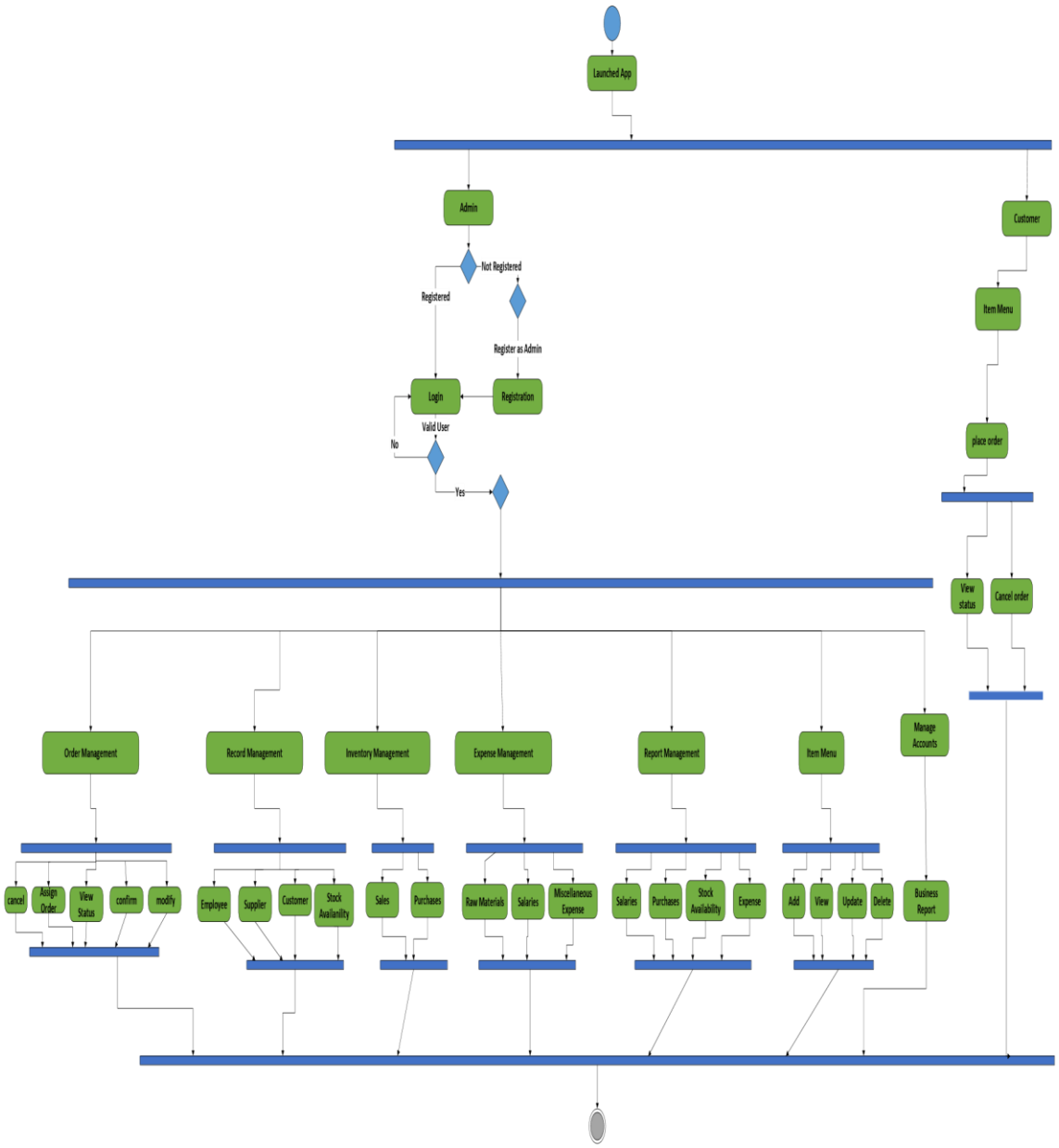


Figure 3.8: State Chart Diagram

4.1.10. Data Model

The data model is a subset of the implementation model, which describes the logical and physical representation of persistent data in the system.

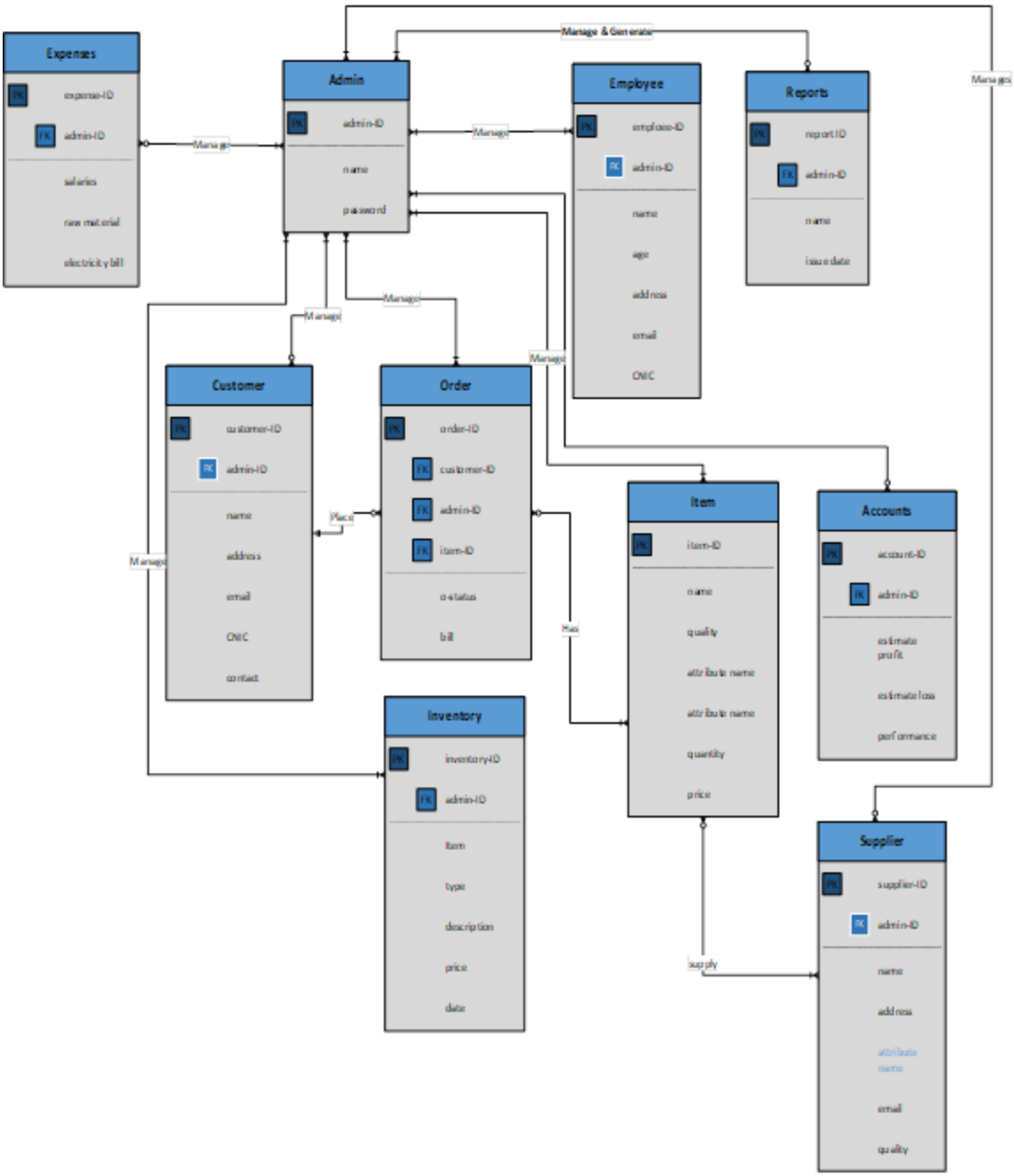


Figure 3.9: Data Model

SQL CODE

```sql
--Database name is
--dbms_project1
--by KHURRAM SHAHZAD (20021519-036)
--   Muhammad Munir (008)
--   Muhammad Faizan (115)
-------------------------------------------------------------------------
--Please go below of this code to read the instruction to run this
--code properly
--------------------------------------------------------------------------
create table General_info(

--General table attributes are here
--This will be used for every table in database

GID int not null ,
First_name varchar(20),
Last_name  varchar(25),
CNIC       varchar(20),
DOB        varchar(20) ,
Gender     varchar(1),
Marital_Status varchar(20),
Address    varchar(20),
Phone_number varchar(20),
job_title varchar(20),
primary key(GID));
--drop table General_info
--I have added all the classes information in one general table

insert into General_info values(1,'khurram','shahzad','34202-2412785-9','12-12-
2000','M','Married','Kotla arab ali khan','3177647269','Manager')
insert into General_info values(2,'Robert ','kiyoski','32302-2412785-9','11-12-
1999','M','Married','America','318494469','Admin')
insert into General_info values(3,'alina ','noor','34202-24755785-9','11-2-
1981','F','Unmarried','Gujrat ','317383269','Manager')
insert into General_info values(4,'Ammarah','Khalid','34202-2414785-9','03-3-
1999','F','married','Sialkot','3144004825','Manager')
insert into General_info values(5,'Shumile','Mirza','24202-34785-9','01-1-
1985','M','Unmarried','Gujrat','337344884','Manager')
insert into General_info values(6,'Munir ','Butt','14202-241237385-1','02-1-
2000','M','married','karnana','238495959','Manager')
insert into General_info values(7,'iftihar','akhtar','02202-2848485-2','03-9-
1960','M','married','malikpur','3184885269','Employee')
insert into General_info values(8,'daud ','kim','23202-234785-9','04-4-
2022','M','Unmarried','sailakot','124847269','Employee')
insert into General_info values(9,'hassan','tahir','32342-2495585-0','05-12-
1991','M','married','lahore','319494433','Employee')
insert into General_info values(10,'waqas','fazal','12202-2414945-7','04-12-
2000','M','married','karachi','313437269','Employee')
insert into General_info values(11,'waheed ','aslam','14202-3412784-5','11-2-
1972','M','Unmarried','korangi','3173343469','Employee')
insert into General_info values(12,'nawaz','khan','33202-23343785-6','06-03-
1999','M','divorced','peshawar','113849333','Manager')
insert into General_info values(13,'fazal','kakad','14202-2595433-7','30-4-
2001','M','Unmarried','kashmir','323848449','Employee')
insert into General_info values(14,'faizan','khakid','33422-25343785-2','3-2-
1992','M','married','faisalabad','31723232269','client')
insert into General_info values(15,'johnny','james','23222-2423285-2','1-1-
1994','M','Unmarried','lahore','31383837269','client')
```

```sql
insert into General_info values(16,'farheen ','nasira','32202-2434385-9','09-3-
2011','F','Unmarried','wazirabad','3134343469','client')
insert into General_info values(17,'usama','alsma','34202-23343435-5','1-2-
1939','M','married','noorpur','3957647269','salesman')
insert into General_info values(18,'waheeda','fasani','23204-2343785-8','9-2-
1900','F','divorced','delhi','3174494439','salesamn')
insert into General_info values(19,'regina','strong','32322-25593785-1','09-1-
2000','F','Unmarried','washington','3343433369','salesalady')
insert into General_info values(20,'kirby','richard','34333-24343785-2','5-3-
2001','M','divorced','lahore','317348333','client')
insert into General_info values(21,'ahmed ','raza','3302-24134385-1','18-2-
2002','M','divorced','pasroor','317443369','client')
insert into General_info values(22,'rachella','robinson','34302-23434385-7','6-2-
1993','M','married','paris','1193833269','client')
insert into General_info values(23,'arlie','shea','33202-23433785-9','12-12-
1990','F','married','hong kong','31773733269','saleslady')
insert into General_info values(24,'loren','byrd','33432-2434385-9','03-3-
1980','M','Unmarried','lahore','313433639','salesman')
insert into General_info values(25,'tracy ','wolf','22202-2333385-3','11-8-
1996','M','married','Barnala','31373732269','salesman')
insert into General_info values(26,'faisal','qureshi','94202-2434785-3','23-2-
1993','M','Unmarried','Daska','31322269','Doctor')
insert into General_info values(27,'mohsin','habib','22202-243232385-9','12-12-
2000','M','Unmarried','johnpur','338393232','Doctor')
insert into General_info values(28,'faizana','syeda','62484-2874325-4','5-3-
1993','F','Divorced','korangi','3858573995','Doctor')
insert into General_info values(29,'imea','falata','53202-2385745-5','11-4-
1973','F','married','fazalpur','35234247269','Doctor')
insert into General_info values(30,'Ghalib','khan','84032-25844785-3','9-3-
2000','M','Unmarried','Chitral','3858939369','Doctor')
insert into General_info values(31,'noror','badiqa','23202-25883785-9','3-3-
1999','F','divorced','jarkhand','318995369','Doctor')
insert into General_info values(32,'harry','tom','20000-23852785-3','03-3-
1990','M','married','lahore','3288447269','Technical staff')
insert into General_info values(33,'faisala','qureshi','34323-2343785-9','1-1-
2000','M','Unmarried','khaniwal','3385767269','technical staff')
insert into General_info values(34,'nazakat','javed','31202-2413455-2','12-7-
1992','M','married','karachi','328959533','technical staff')
insert into General_info values(35,'ayesha','javaria','14202-24534785-7','8-12-
1990','F','married','bahawalpur','8857393232','technical staff')
insert into General_info values(36,'xavier','zain','34322-24543785-1','2-9-
2001','M','divorced','jadianwala','8573994595','technical staff')

-------------------------------------------------------------
select *from General_info;                                  --
-------------------------------------------------------------
-- I have used dintinct here to get unique values          --

select distinct First_name from General_info;               --
--I have used ascending and descending here                 --
select* from General_info order by DOB;                     --
select*from General_info order by GID asc;                  --
select*from General_info order by  GID desc;                --

-------------------------------------------------------------

--create the table for admin use
--I have created table no.1 here
```

```
create table Admin_table1(
Admin_id int not null,
salary int ,
experinence_before int ,
worked_for varchar(25),
primary key(Admin_id),
GID int ,
--on delete and on update cascade will delete or update the child table with the change
of parent table
-- I have used the triggers here
foreign key (GID) references General_info(GID) on delete cascade on update cascade);
--drop table Admin_table1
--data of admin
insert into Admin_table1 values(90,200000,19,'NA',2)
-----------------------------------------------------------------
--commands to run the admin table information                  --
-----------------------------------------------------------------
select *from General_info;                                      --
select *from Admin_table1;
select General_info.GID,
General_info.First_name,General_info.Last_name,Admin_table1.salary,Admin_table1.experinen
ce_before,
Admin_table1.worked_for
from General_info inner join Admin_table1
on General_info.GID=Admin_table1.Admin_id;                            --
-----------------------------------------------------------------

--I have created the table for manager data here
--I have create the no.2 table here

create table Manager_table(
Manager_id int not null,
working_in varchar(25),
worked_for varchar(25),
salary float,
worked_under varchar(20),
primary key(Manager_id),
GID int ,
Admin_id int,
foreign key (GID) references general_info (GID) on delete cascade on update cascade,
foreign key (Admin_id ) references Admin_table1( Admin_id ));
--drop table Manager_table
insert into Manager_table values(12,'burraq pharmacy','itfaq pharmacy',128494,'robert
kiyoski',1,11)
insert into Manager_table values(13,'itfaq pharmacy','burraq pharmacy',128494,'robert
kiyoski',4,11)
insert into Manager_table values(14,'pure pharmacy','shoot pharmacy',142494,'robert
kiyoski',3,11)
insert into Manager_table values(15,'liaza pharmacy','pure pharmacy',1754494,'robert
kiyoski',6,11)
insert into Manager_table values(16,'horbon pharmacy','shareef pharmacy',1434494,'robert
kiyoski',12,11)
insert into Manager_table values(17,'Noor pharmacy','kareem pharmacy',12844,'robert
kiyoski',5,11)
-----------------------------------------------------------------
--aggregat functions has been used here                         --
--
```

```
-----------------------------------------------------------------
select sum(salary) as all_managers_pay from Manager_table;    --
select max(salary) as highest_pay from Manager_table;           --
select min(salary) as minimum_pay from Manager_table;           --
select avg(salary) as Average_salary from Manager_table;        --
select count(salary ) as total_managers from Manager_table;     --


                    --
-----------------------------------------------------------------
--commands to run the managers code
-----------------------------------------------------------------
select*from  General_info;                                      --
select*from  Admin_table1;                                      --
select *from Manager_table;

--with join
select
General_info.First_name,General_info.Last_name,Manager_table.worked_for,Manager_table.sal
ary
from General_info inner join Manager_table
on General_info.GID=Manager_table.GID;                       --
-----------------------------------------------------------------

-- I have created the table for the employees here
--I have created the no.3 table here

create table Employess(
Employee_id int ,
working_in  varchar(20),
worked_for  varchar(25),
salary      float,
experience int,
working_under varchar(20),
Manager_id int,
GID int ,
primary key(Employee_id),
foreign key(GID) references general_info(GID) on delete cascade on update cascade,

foreign key (Manager_id) references Manager_table (Manager_id) );
--drop table Employess
insert into Employess values(21,'burraq pharmacy','itfaq pharmacy',12393,3,'Ammarah
Khalid',13,7)
insert into Employess values(22,'itfaq pharmacy','burraq pharmacy',12373,4,'Khurram
Shahzad',12,8)
insert into Employess values(23,'pure pharmacy','shoot pharmacy',127464,12,'alina
noor',14,9)
insert into Employess values(24,'liaza pharmacy','pure pharmacy',122823,4,'Shumile
mirza',17,10)
insert into Employess values(25,'horbon pharmacy','shareef pharmacy',148553,8,'munir
butt',15,11)
insert into Employess values(26,'noor pharmacy','kareem pharmacy',200023,10,'nawaz
khan',16,13)
--------------------------------------------------------------------------------
-- I have used where and , between here                                       --
select salary from Employess where salary between 20000 and 200000;           --
                                                                              --
--------------------------------------------------------------------------------
```

```
--commands for the employee table to run it.                                --

                              --
-------------------------------------------------------------------------------

                                          --
select*from General_info;                                                   --
select*from Manager_table;                                                  --
select*from Employess;                                                      --

                                          --
-----------------------------------------------------------------------------
-- I have created the table no.4 here
--I have created this table for the client data

create table client_table(
client_id int,
ordered_medicine varchar(20),
order_price varchar(20),
add_to_cart varchar(20),
total_buying int,
buying_pharmacy varchar(20),
linked_doctor varchar(20),

disease_to_patient varchar(20),
primary key (client_id),
GID int ,
foreign key (GID) references general_info (GID) on delete cascade on update cascade
);
--drop table client_table
--at last i added the general tabele id to search complete data
insert into client_table values(31,'panadol',320,'lipiget',1000,'burraq pharmacy','faisal
qureshi','Headache',15)
insert into client_table values(32,'lipiget',340,'panadol',2000,'itfaq pharmacy' ,'ghalib
khan','blood sugar',16)
insert into client_table values(33,'asphil',360,'lipirex',3000,'noor pharmacy','noror
badavia','fever',14)
insert into client_table values(34,'getryl',380,'asphil',4000,'liaza pharmacy','faizan
syeda','TB',20)
insert into client_table values(35,'calcuim-c',400,'getryl',5000,'pure pharmacy','mohsin
habib','diearah',21)
insert into client_table values(36,'lipirex',420,'calcuim-C',6000,'horbon pharmacy','imea
falata','heart problem',22);

--------------------------------------------------------------------------
-- basic commands to run the client table data
--

                                  --
select *from client_table;                                              --
select*from General_info;
select General_info.First_name,General_info.Last_name,
client_table.ordered_medicine,client_table.order_price,client_table.add_to_cart,
client_table.total_buying,client_table.buying_pharmacy,
client_table.linked_doctor,client_table.disease_to_patient
from General_info
inner join client_table on General_info.GID=client_table.GID;
                                                                        --
```

```
                                    --
--I have used the constraints here
        --

--

                                    --

                                    --

                                    --
-----------------------------------------------------------------------
-- I have created the doctors table here to store the data of doctors
-- The table no.5 is here.


create table Doctors_table (
doctors_id  int unique ,
linked_pharmacy varchar(20),
experience int,
qualified_in varchar(30),
client_id int ,
GID int ,
primary key(doctors_id),
foreign key (GID) references General_info(GID) on delete cascade on update cascade,
foreign key (client_id) references client_table (client_id)) ;
--drop table Doctors_table
insert into Doctors_table values(41,'burraq pharmacy',10,'neurosurgeon',31,26)
insert into Doctors_table values(42,'itfaq pharmacy',12,'orthopedex',32,30)
insert into Doctors_table values(43,'noor  pharmacy',5,'ENT specialist',33,31)
insert into Doctors_table values(44,'liaza pharmacy',1,'Children specialist',34,28)
insert into Doctors_table values(45,'pure  pharmacy',23,'getrospecialist',35,27)
insert into Doctors_table values(46,'horbon pharmacy',6,'psychologist',36,29);

--------------------------------------------------------------------------
--Basic command to run the doctors table code
       --

                                         --

                                         --
select*from Doctors_table;
                    --
select*from client_table;
                    --
select*from General_info;
                    --

                                         --

                                         --

                                         --

                                         --
-------------------------------------------------------------------------
-- this is a table to store the salesmen data
```

```
-- the no.6 table is here

create table Salesmen_table(
Salesmen_id int ,
stock_sold varchar(20),
working_in  varchar(20),
worked_for  varchar(25),
earned_commission float,
withdrew_commission float,
worked_years int,
working_under varchar(20),
GID int ,
Manager_id int,
primary key(Salesmen_id),
foreign key(GID) references general_info(GID) ,
foreign key(Manager_id) references Manager_table (Manager_id));
--drop table Salesmen_table
insert into Salesmen_table values(51,'panadol','burraq pharmacy','itfaq pharmacy'
,5000,1000,2,'Khurram Shahzad',17,12 )
insert into Salesmen_table values(52,'lipiget','itfaq pharmacy','burraq pharma'
,6000,2000,3,'Ammarah Khalid', 18, 13)
insert into Salesmen_table values(53,'lipirex','pure pharmacy','shoot pharmacy'
,7000,3000,4,'Alina Noor', 19, 14)
insert into Salesmen_table values(54,'getryl','liaza pharmacy','pure pharmacy'
,8000,4000,5,'Munir Butt',23 ,15 )
insert into Salesmen_table values(55,'asphil','liaza pharmacy','pure pharmacy'
,9000,5000,6,'Munir Butt', 24, 15)
insert into Salesmen_table values(56,'Calcuim-C','horbon pharmacy','shareef pharmacy'
,10000,6000,7,'Nawaz Khan',25 , 16)


-----------------------------------------------------------------------------------------
-
--Basic commands for to run the salemen table

              --
                              --
select*from Salesmen_table;

select General_info.First_name,General_info.Last_name,
General_info.Gender,Salesmen_table.Salesmen_id,Salesmen_table.stock_sold,Salesmen_table.w
orked_for,
Salesmen_table.earned_commission,Salesmen_table.withdrew_commission,
Salesmen_table.worked_years,Salesmen_table.working_under,
Salesmen_table.Manager_id
from General_info inner join
Salesmen_table on General_info.GID=Salesmen_table.Manager_id;
              --

                                                                  --

                                                                  --

                                                                  --

                                                                  --

                                                                  --
```

```
-----------------------------------------------------------------------------------------
--I have created the table for inventory
-- this is no.7 table in our sql
create table inventory_table(
inventory_id int,
Medicine_name varchar(20),
medicine_price varchar(20),
destination_pharmacy varchar(20),
sold_by_employee varchar(20),
salesmen_connected varchar(20),
stock_sold_out int,
stock_available int,
source_of_medicine varchar(20),
expiry_date Varchar(10),
primary key(inventory_id),
Employee_id int,
Salesmen_id int,


foreign key(Employee_id) references Employess(Employee_id)on delete cascade on update
cascade ,
foreign key(Salesmen_id) references Salesmen_table(Salesmen_id));
--drop table inventory_table

insert into inventory_table values(61,'panadol',60,'burraq pharmacy','johnny
james','usama aslam',
23,100,'al-shareef pharma','31-12-2023',21,51)
insert into inventory_table values(62,'lipiget',70,'itfaq pharmacy','daud kim','waheed
fasani',
24,200,'careem pharma','31-1-2024',22,52)
insert into inventory_table values(63,'lipirex',80,'horbon pharmacy','waheed
aslam','tracy wolf',
25,300,'safina  pharma','12-12-2026',25,56)
insert into inventory_table values(64,'getryl',90,'noor pharmacy','fazal kakad','NA',
26,400,'barqat pharma','9-05-2023',26,51)
insert into inventory_table values(65,'calcium-c',100,'liaza pharmacy','waqas
fazal','loren byrd',
27,500,'nupuria pharma','1-7-2023',24,54)
insert into inventory_table values(66,'asphil',110,'pure pharmacy','hassan tahir','tracy
wolf',
28,600,'atgate pharma','12-2-2020',23,53)

-----------------------------------------------------------------------------------------
---

select*from inventory_table;


-----------------------------------------------------------------------------------------
-----
-- I have created the table for the technical staff here
-- this is no.8 table in our sql code
create table tech_staff_table(
tech_staff_id int,
working_in varchar(20),
working_under varchar(30),
salary int ,
GID int,
```

```
Manger_id int,
primary key(tech_staff_id),
foreign key(GID) references General_info(GID));

--drop table tech_staff_table
insert into tech_staff_table values (71,'burraq pharmacy','khurram shahzad',15000,32 ,1)
insert into tech_staff_table values (72,'itfaq pharmacy','Ammarah khalid',20000,33 ,4)
insert into tech_staff_table values (73,'pure pharmacy','Alina noor ',25000, 34,3)
insert into tech_staff_table values (74,'liaza pharmacy','Munir Butt',30000,35 ,6)
insert into tech_staff_table values (75,'noor pharmacy','Shumile Mirza',35000,36 ,5)
select*from tech_staff_table




---------------------------- views for the system users--------------------------------
--------
--I have created a view for the users here to see general information
--drop view all_personal_data
create view all_personal_data as
select First_Name,Last_Name,CNIC,DOB,Gender,Marital_Status,Address,Phone_number,job_title
from General_info;
select *from  all_personal_data

------------------------------------------------------------------------------------------
------------



---------------------------------------------------- Logins for the users of database
system---------------------
--I have created the login for each user here
-- By this only authentic user will interact to system
-- I have stored all the data for the login here
create table login(
username varchar(30),
password varchar(10),
primary key(username),
GID int,
foreign key(GID) references General_info (GID));
insert into login values('khurram1','code',1)
insert into login values('robert2','code',2)
insert into login values('alina3','code',3)
insert into login values('ammarah4','code',4)
insert into login values('shumile5','code',5)
```

```sql
insert into login values('munir6','code',6)
insert into login values('iftihar7','code',7)
insert into login values('daud8','code',8)
insert into login values('hassan9','code',9)
insert into login values('waqas10','code',10)
insert into login values('waheed11','code',11)
insert into login values('nawaz12','code',12)
insert into login values('fazal13','code',13)
insert into login values('faizan14','code',14)
insert into login values('johnny15','code',15)
insert into login values('farheen16','code',16)
insert into login values('usama17','code',17)
insert into login values('waheed18','code',18)
insert into login values('regina19','code',19)
insert into login values('kirby20','code',20)
insert into login values('ahmed21','code',21)
insert into login values('rachella22','code',22)
insert into login values('arlie23','code',23)
insert into login values('loren24','code',24)
insert into login values('tracy25','code',25)
insert into login values('faisal26','code',26)
insert into login values('mohsin27','code',27)
insert into login values('faizana28','code',28)
insert into login values('imea29','code',29)
insert into login values('ghalib30','code',30)
insert into login values('noror31','code',31)
insert into login values('harry32','code',32)
insert into login values('faisala33','code',33)
insert into login values('nazakat34','code',34)
insert into login values('ayesha35','code',35)
insert into login values('xavier36','code',36)
-----------------------------------------------
select*from login order by GID

-----------------------------------------------
--admin login
create login robert2 with password='code'
create user robert2 for login robert2

--managers login
create login khurram1 with password='code'
create user khurram1 for login khurram1

create login ammarah4 with password='code'
create user ammarah4 for login ammarah4

create login alina3 with password='code'
create user alina3 for login alina3

create login shumile5 with password='code'
create user shumile5 for login shumile5

create login munir6 with password='code'
create user munir6 for login munir6

create login nawaz12 with password='code'
create user nawaz12 for login nawaz12
```

```sql
-- login for the employees
create login iftikhar7 with password='code'
create user iftikhar7 for login iftikhar7

create login daud8 with password='code'
create user daud8 for login daud8

create login hassan9 with password='code'
create user hassan9 for login hassan9

create login waqas10 with password='code'
create user waqas10 for login waqas10

create login waheed11 with password='code'
create user waheed11 for login waheed11

create login fazal13 with password='code'
create user fazal13 for login fazal13

--login for the clients

create login faizan14 with password='code'
create user faizan14 for login faizan14

create login johnny15 with password='code'
create user johnny15 for login johnny15

create login farheen16 with password='code'
create user farheen16 for login farheen16

create login kirby20 with password='code'
create user kirby20 for login kirby20

create login ahemd21 with password='code'
create user ahmed21 for login ahemd21

create login rachella22 with password='code'
create user rachella22 for login rachella22

--login for the doctors

create login faisal26 with password='code'
create user faisal26 for login faisal26

create login mohsin27 with password='code'
create user mohsin27 for login mohsin27

create login faizana28 with password='code'
create user faizana28 for login faizana28

create login imea29 with password='code'
create user imea29 for login imea29

create login ghalib30 with password='code'
create user ghalib30 for login ghalib30

create login noror31 with password='code'
create user noror31 for login noror31
```

```sql
--login for the salesmen and saleladies

create login usama17 with password='code'
create user usama17 for login usama17

create login waheed18 with password='code'
create user waheed18 for login waheed18

create login regina19 with password='code'
create user regina19 for login regina19

create login arlie23 with password='code'
create user arlie23 for login arlie23

create login loren24 with password='code'
create user loren24 for login loren24

create login tracy25 with password='code'
create user tracy25 for login tracy25

--login for the Techincal staff is here

create login harry32 with password='code'
create user harry32 for login harry32


create login faisala33 with password='code'
create user faisala33 for login faisala33

create login nazakat34 with password='code'
create user nazakat34 for login nazakat34

create login ayesha35 with password='code'
create user ayesha35 for login ayesha35

create login xavier36 with password='code'
create user xavier36 for login xavier36

--Database name is
--dbms_project1
-------------------------------------------------------------------------
--Please go below of this code to read the instruction to run this
--code properly
---------------------------------------------------------------------------
create table General_info(

--General table attributes are here
--This will be used for every table in database

GID int not null ,
First_name varchar(20),
Last_name  varchar(25),
CNIC       varchar(20),
DOB        varchar(20) ,
Gender     varchar(1),
Marital_Status varchar(20),
Address    varchar(20),
```

```
Phone_number varchar(20),
job_title varchar(20),
primary key(GID));
--drop table General_info
--I have added all the classes information in one general table

insert into General_info values(1,'khurram','shahzad','34202-2412785-9','12-12-
2000','M','Married','Kotla arab ali khan','3177647269','Manager')
insert into General_info values(2,'Robert ','kiyoski','32302-2412785-9','11-12-
1999','M','Married','America','318494469','Admin')
insert into General_info values(3,'alina ','noor','34202-24755785-9','11-2-
1981','F','Unmarried','Gujrat ','317383269','Manager')
insert into General_info values(4,'Ammarah','Khalid','34202-2414785-9','03-3-
1999','F','married','Sialkot','3144004825','Manager')
insert into General_info values(5,'Shumile','Mirza','24202-34785-9','01-1-
1985','M','Unmarried','Gujrat','337344884','Manager')
insert into General_info values(6,'Munir ','Butt','14202-241237385-1','02-1-
2000','M','married','karnana','238495959','Manager')
insert into General_info values(7,'iftihar','akhtar','02202-2848485-2','03-9-
1960','M','married','malikpur','3184885269','Employee')
insert into General_info values(8,'daud ','kim','23202-234785-9','04-4-
2022','M','Unmarried','sailakot','124847269','Employee')
insert into General_info values(9,'hassan','tahir','32342-2495585-0','05-12-
1991','M','married','lahore','319494433','Employee')
insert into General_info values(10,'waqas','fazal','12202-2414945-7','04-12-
2000','M','married','karachi','313437269','Employee')
insert into General_info values(11,'waheed ','aslam','14202-3412784-5','11-2-
1972','M','Unmarried','korangi','3173343469','Employee')
insert into General_info values(12,'nawaz','khan','33202-23343785-6','06-03-
1999','M','divorced','peshawar','113849333','Manager')
insert into General_info values(13,'fazal','kakad','14202-2595433-7','30-4-
2001','M','Unmarried','kashmir','323848449','Employee')
insert into General_info values(14,'faizan','khakid','33422-25343785-2','3-2-
1992','M','married','faisalabad','31723232269','client')
insert into General_info values(15,'johnny','james','23222-2423285-2','1-1-
1994','M','Unmarried','lahore','31383837269','client')
insert into General_info values(16,'farheen ','nasira','32202-2434385-9','09-3-
2011','F','Unmarried','wazirabad','3134343469','client')
insert into General_info values(17,'usama','alsma','34202-23343435-5','1-2-
1939','M','married','noorpur','3957647269','salesman')
insert into General_info values(18,'waheeda','fasani','23204-2343785-8','9-2-
1900','F','divorced','delhi','3174494439','salesamn')
insert into General_info values(19,'regina','strong','32322-25593785-1','09-1-
2000','F','Unmarried','washington','3343433369','salesalady')
insert into General_info values(20,'kirby','richard','34333-24343785-2','5-3-
2001','M','divorced','lahore','317348333','client')
insert into General_info values(21,'ahmed ','raza','3302-24134385-1','18-2-
2002','M','divorced','pasroor','317443369','client')
insert into General_info values(22,'rachella','robinson','34302-23434385-7','6-2-
1993','M','married','paris','1193833269','client')
insert into General_info values(23,'arlie','shea','33202-23433785-9','12-12-
1990','F','married','hong kong','31773733269','saleslady')
insert into General_info values(24,'loren','byrd','33432-2434385-9','03-3-
1980','M','Unmarried','lahore','313433639','salesman')
insert into General_info values(25,'tracy ','wolf','22202-2333385-3','11-8-
1996','M','married','Barnala','31373732269','salesman')
insert into General_info values(26,'faisal','qureshi','94202-2434785-3','23-2-
1993','M','Unmarried','Daska','31322269','Doctor')
```

```
insert into General_info values(27,'mohsin','habib','22202-243232385-9','12-12-
2000','M','Unmarried','johnpur','338393232','Doctor')
insert into General_info values(28,'faizana','syeda','62484-2874325-4','5-3-
1993','F','Divorced','korangi','3858573995','Doctor')
insert into General_info values(29,'imea','falata','53202-2385745-5','11-4-
1973','F','married','fazalpur','35234247269','Doctor')
insert into General_info values(30,'Ghalib','khan','84032-25844785-3','9-3-
2000','M','Unmarried','Chitral','3858939369','Doctor')
insert into General_info values(31,'noror','badiqa','23202-25883785-9','3-3-
1999','F','divorced','jarkhand','318995369','Doctor')
insert into General_info values(32,'harry','tom','20000-23852785-3','03-3-
1990','M','married','lahore','3288447269','Technical staff')
insert into General_info values(33,'faisala','qureshi','34323-2343785-9','1-1-
2000','M','Unmarried','khaniwal','3385767269','technical staff')
insert into General_info values(34,'nazakat','javed','31202-2413455-2','12-7-
1992','M','married','karachi','328959533','technical staff')
insert into General_info values(35,'ayesha','javaria','14202-24534785-7','8-12-
1990','F','married','bahawalpur','8857393232','technical staff')
insert into General_info values(36,'xavier','zain','34322-24543785-1','2-9-
2001','M','divorced','jadianwala','8573994595','technical staff')

------------------------------------------------------------
select *from General_info;                                --
------------------------------------------------------------
-- I have used dintinct here to get unique values         --

select distinct First_name from General_info;             --
--I have used ascending and descending here               --
select* from General_info order by DOB;                   --
select*from General_info order by GID asc;                --
select*from General_info order by  GID desc;                --

------------------------------------------------------------

--create the table for admin use
--I have created table no.1 here

create table Admin_table1(
Admin_id int not null,
salary int ,
experinence_before int ,
worked_for varchar(25),
primary key(Admin_id),
GID int ,
--on delete and on update cascade will delete or update the child table with the change
of parent table
-- I have used the triggers here
foreign key (GID) references General_info(GID) on delete cascade on update cascade);
--drop table Admin_table1
--data of admin
insert into Admin_table1 values(90,200000,19,'NA',2)
------------------------------------------------------------
--commands to run the admin table information                --
------------------------------------------------------------
select *from General_info;                                  --
select *from Admin_table1;
```

```
select General_info.GID,
General_info.First_name,General_info.Last_name,Admin_table1.salary,Admin_table1.experinen
ce_before,
Admin_table1.worked_for
from General_info inner join Admin_table1
on General_info.GID=Admin_table1.Admin_id;                          --
-----------------------------------------------------------------

--I have created the table for manager data here
--I have create the no.2 table here

create table Manager_table(
Manager_id int not null,
working_in varchar(25),
worked_for varchar(25),
salary float,
worked_under varchar(20),
primary key(Manager_id),
GID int ,
Admin_id int,
foreign key (GID) references general_info (GID) on delete cascade on update cascade,
foreign key (Admin_id ) references Admin_table1( Admin_id ));
--drop table Manager_table
insert into Manager_table values(12,'burraq pharmacy','itfaq pharmacy',128494,'robert
kiyoski',1,11)
insert into Manager_table values(13,'itfaq pharmacy','burraq pharmacy',128494,'robert
kiyoski',4,11)
insert into Manager_table values(14,'pure pharmacy','shoot pharmacy',142494,'robert
kiyoski',3,11)
insert into Manager_table values(15,'liaza pharmacy','pure pharmacy',1754494,'robert
kiyoski',6,11)
insert into Manager_table values(16,'horbon pharmacy','shareef pharmacy',1434494,'robert
kiyoski',12,11)
insert into Manager_table values(17,'Noor pharmacy','kareem pharmacy',12844,'robert
kiyoski',5,11)
-----------------------------------------------------------------
--aggregat functions has been used here                          --
--
-----------------------------------------------------------------
select sum(salary) as all_managers_pay from Manager_table;    --
select max(salary) as highest_pay from Manager_table;           --
select min(salary) as minimum_pay from Manager_table;           --
select avg(salary) as Average_salary from Manager_table;        --
select count(salary ) as total_managers from Manager_table;     --


                    --
-----------------------------------------------------------------
--commands to run the managers code
-----------------------------------------------------------------
select*from  General_info;                                      --
select*from  Admin_table1;                                      --
select *from Manager_table;

--with join
select
General_info.First_name,General_info.Last_name,Manager_table.worked_for,Manager_table.sal
ary
```

```
from General_info inner join Manager_table
on General_info.GID=Manager_table.GID;                        --
-----------------------------------------------------------------

-- I have created the table for the employees here
--I have created the no.3 table here

create table Employess(
Employee_id int ,
working_in  varchar(20),
worked_for  varchar(25),
salary      float,
experience int,
working_under varchar(20),
Manager_id int,
GID int ,
primary key(Employee_id),
foreign key(GID) references general_info(GID) on delete cascade on update cascade,

foreign key (Manager_id) references Manager_table (Manager_id) );
--drop table Employess
insert into Employess values(21,'burraq pharmacy','itfaq pharmacy',12393,3,'Ammarah
Khalid',13,7)
insert into Employess values(22,'itfaq pharmacy','burraq pharmacy',12373,4,'Khurram
Shahzad',12,8)
insert into Employess values(23,'pure pharmacy','shoot pharmacy',127464,12,'alina
noor',14,9)
insert into Employess values(24,'liaza pharmacy','pure pharmacy',122823,4,'Shumile
mirza',17,10)
insert into Employess values(25,'horbon pharmacy','shareef pharmacy',148553,8,'munir
butt',15,11)
insert into Employess values(26,'noor pharmacy','kareem pharmacy',200023,10,'nawaz
khan',16,13)
-------------------------------------------------------------------------------
-- I have used where and , between here                                      --
select salary from Employess where salary between 20000 and 200000;           --
                                                                              --
-------------------------------------------------------------------------------
--commands for the employee table to run it.                                  --

                                --
-------------------------------------------------------------------------------

                                            --
select*from General_info;                                                     --
select*from Manager_table;                                                    --
select*from Employess;                                                        --

                                            --
-------------------------------------------------------------------------------
-- I have created the table no.4 here
--I have created this table for the client data

create table client_table(
client_id int,
ordered_medicine varchar(20),
order_price varchar(20),
add_to_cart varchar(20),
```

```
total_buying int,
buying_pharmacy varchar(20),
linked_doctor varchar(20),

disease_to_patient varchar(20),
primary key (client_id),
GID int ,
foreign key (GID) references general_info (GID) on delete cascade on update cascade
);
--drop table client_table
--at last i added the general tabele id to search complete data
insert into client_table values(31,'panadol',320,'lipiget',1000,'burraq pharmacy','faisal
qureshi','Headache',15)
insert into client_table values(32,'lipiget',340,'panadol',2000,'itfaq pharmacy' ,'ghalib
khan','blood sugar',16)
insert into client_table values(33,'asphil',360,'lipirex',3000,'noor pharmacy','noror
badavia','fever',14)
insert into client_table values(34,'getryl',380,'asphil',4000,'liaza pharmacy','faizan
syeda','TB',20)
insert into client_table values(35,'calcuim-c',400,'getryl',5000,'pure pharmacy','mohsin
habib','diearah',21)
insert into client_table values(36,'lipirex',420,'calcuim-C',6000,'horbon pharmacy','imea
falata','heart problem',22);

-----------------------------------------------------------------------------
-- basic commands to run the client table data
--

                                    --
select *from client_table;                                                --
select*from General_info;
select General_info.First_name,General_info.Last_name,
client_table.ordered_medicine,client_table.order_price,client_table.add_to_cart,
client_table.total_buying,client_table.buying_pharmacy,
client_table.linked_doctor,client_table.disease_to_patient
from General_info
inner join client_table on General_info.GID=client_table.GID;
                                                                      --

                                    --
--I have used the constraints here
        --

--

                                    --

                                    --

                                    --
-----------------------------------------------------------------------------
-- I have created the doctors table here to store the data of doctors
-- The table no.5 is here.


create table Doctors_table (
doctors_id  int unique ,
linked_pharmacy varchar(20),
```

```
experience int,
qualified_in varchar(30),
client_id int ,
GID int ,
primary key(doctors_id),
foreign key (GID) references General_info(GID) on delete cascade on update cascade,
foreign key (client_id) references client_table (client_id)) ;
--drop table Doctors_table
insert into Doctors_table values(41,'burraq pharmacy',10,'neurosurgeon',31,26)
insert into Doctors_table values(42,'itfaq pharmacy',12,'orthopedex',32,30)
insert into Doctors_table values(43,'noor  pharmacy',5,'ENT specialist',33,31)
insert into Doctors_table values(44,'liaza pharmacy',1,'Children specialist',34,28)
insert into Doctors_table values(45,'pure  pharmacy',23,'getrospecialist',35,27)
insert into Doctors_table values(46,'horbon pharmacy',6,'psychologist',36,29);

-----------------------------------------------------------------------------
--Basic command to run the doctors table code
       --

                                        --

                                        --
select*from Doctors_table;
                     --
select*from client_table;
                     --
select*from General_info;
                     --

                                        --

                                        --

                                        --

                                        --
-----------------------------------------------------------------------------
-- this is a table to store the salesmen data
-- the no.6 table is here

create table Salesmen_table(
Salesmen_id int ,
stock_sold varchar(20),
working_in  varchar(20),
worked_for  varchar(25),
earned_commission float,
withdrew_commission float,
worked_years int,
working_under varchar(20),
GID int ,
Manager_id int,
primary key(Salesmen_id),
foreign key(GID) references general_info(GID) ,
foreign key(Manager_id) references Manager_table (Manager_id));
--drop table Salesmen_table
insert into Salesmen_table values(51,'panadol','burraq pharmacy','itfaq pharmacy'
,5000,1000,2,'Khurram Shahzad',17,12 )
insert into Salesmen_table values(52,'lipiget','itfaq pharmacy','burraq pharma'
```

```
,6000,2000,3,'Ammarah Khalid', 18, 13)
insert into Salesmen_table values(53,'lipirex','pure pharmacy','shoot pharmacy'
,7000,3000,4,'Alina Noor', 19, 14)
insert into Salesmen_table values(54,'getryl','liaza pharmacy','pure pharmacy'
,8000,4000,5,'Munir Butt',23 ,15 )
insert into Salesmen_table values(55,'asphil','liaza pharmacy','pure pharmacy'
,9000,5000,6,'Munir Butt', 24, 15)
insert into Salesmen_table values(56,'Calcuim-C','horbon pharmacy','shareef pharmacy'
,10000,6000,7,'Nawaz Khan',25 , 16)


-----------------------------------------------------------------------------------------
-
--Basic commands for to run the salemen table

              --
                            --
select*from Salesmen_table;

select General_info.First_name,General_info.Last_name,
General_info.Gender,Salesmen_table.Salesmen_id,Salesmen_table.stock_sold,Salesmen_table.w
orked_for,
Salesmen_table.earned_commission,Salesmen_table.withdrew_commission,
Salesmen_table.worked_years,Salesmen_table.working_under,
Salesmen_table.Manager_id
from General_info inner join
Salesmen_table on General_info.GID=Salesmen_table.Manager_id;
              --

                                                                --

                                                                --

                                                                --

                                                                --

                                                                --
-----------------------------------------------------------------------------------------
--I have created the table for inventory
-- this is no.7 table in our sql
create table inventory_table(
inventory_id int,
Medicine_name varchar(20),
medicine_price varchar(20),
destination_pharmacy varchar(20),
sold_by_employee varchar(20),
salesmen_connected varchar(20),
stock_sold_out int,
stock_available int,
source_of_medicine varchar(20),
expiry_date Varchar(10),
primary key(inventory_id),
Employee_id int,
Salesmen_id int,
```

```
foreign key(Employee_id) references Employess(Employee_id)on delete cascade on update
cascade ,
foreign key(Salesmen_id) references Salesmen_table(Salesmen_id));
--drop table inventory_table

insert into inventory_table values(61,'panadol',60,'burraq pharmacy','johnny
james','usama aslam',
23,100,'al-shareef pharma','31-12-2023',21,51)
insert into inventory_table values(62,'lipiget',70,'itfaq pharmacy','daud kim','waheed
fasani',
24,200,'careem pharma','31-1-2024',22,52)
insert into inventory_table values(63,'lipirex',80,'horbon pharmacy','waheed
aslam','tracy wolf',
25,300,'safina  pharma','12-12-2026',25,56)
insert into inventory_table values(64,'getryl',90,'noor pharmacy','fazal kakad','NA',
26,400,'barqat pharma','9-05-2023',26,51)
insert into inventory_table values(65,'calcium-c',100,'liaza pharmacy','waqas
fazal','loren byrd',
27,500,'nupuria pharma','1-7-2023',24,54)
insert into inventory_table values(66,'asphil',110,'pure pharmacy','hassan tahir','tracy
wolf',
28,600,'atgate pharma','12-2-2020',23,53)

-----------------------------------------------------------------------------------------
---

select*from inventory_table;


-----------------------------------------------------------------------------------------
-----
-- I have created the table for the technical staff here
-- this is no.8 table in our sql code
create table tech_staff_table(
tech_staff_id int,
working_in varchar(20),
working_under varchar(30),
salary int ,
GID int,
Manger_id int,
primary key(tech_staff_id),
foreign key(GID) references General_info(GID));

--drop table tech_staff_table
insert into tech_staff_table values (71,'burraq pharmacy','khurram shahzad',15000,32 ,1)
insert into tech_staff_table values (72,'itfaq pharmacy','Ammarah khalid',20000,33 ,4)
insert into tech_staff_table values (73,'pure pharmacy','Alina noor ',25000, 34,3)
insert into tech_staff_table values (74,'liaza pharmacy','Munir Butt',30000,35 ,6)
insert into tech_staff_table values (75,'noor pharmacy','Shumile Mirza',35000,36 ,5)
select*from tech_staff_table





---------------------------- views for the system users--------------------------------
--------
```

```sql
--I have created a view for the users here to see general information
--drop view all_personal_data
create view all_personal_data as
select First_Name,Last_Name,CNIC,DOB,Gender,Marital_Status,Address,Phone_number,job_title
from General_info;
select *from  all_personal_data

-----------------------------------------------------------------------------------------
------------
```

```sql
---------------------------------------------------- Logins for the users of database
system---------------------
--I have created the login for each user here
-- By this only authentic user will interact to system
-- I have stored all the data for the login here
create table login(
username varchar(30),
password varchar(10),
primary key(username),
GID int,
foreign key(GID) references General_info (GID));
insert into login values('khurram1','code',1)
insert into login values('robert2','code',2)
insert into login values('alina3','code',3)
insert into login values('ammarah4','code',4)
insert into login values('shumile5','code',5)
insert into login values('munir6','code',6)
insert into login values('iftihar7','code',7)
insert into login values('daud8','code',8)
insert into login values('hassan9','code',9)
insert into login values('waqas10','code',10)
insert into login values('waheed11','code',11)
insert into login values('nawaz12','code',12)
insert into login values('fazal13','code',13)
insert into login values('faizan14','code',14)
insert into login values('johnny15','code',15)
insert into login values('farheen16','code',16)
insert into login values('usama17','code',17)
insert into login values('waheed18','code',18)
insert into login values('regina19','code',19)
insert into login values('kirby20','code',20)
insert into login values('ahmed21','code',21)
insert into login values('rachella22','code',22)
insert into login values('arlie23','code',23)
insert into login values('loren24','code',24)
```

```sql
insert into login values('tracy25','code',25)
insert into login values('faisal26','code',26)
insert into login values('mohsin27','code',27)
insert into login values('faizana28','code',28)
insert into login values('imea29','code',29)
insert into login values('ghalib30','code',30)
insert into login values('noror31','code',31)
insert into login values('harry32','code',32)
insert into login values('faisala33','code',33)
insert into login values('nazakat34','code',34)
insert into login values('ayesha35','code',35)
insert into login values('xavier36','code',36)
-----------------------------------------------
select*from login order by GID

-----------------------------------------------
--admin login
create login robert2 with password='code'
create user robert2 for login robert2

--managers login
create login khurram1 with password='code'
create user khurram1 for login khurram1

create login ammarah4 with password='code'
create user ammarah4 for login ammarah4

create login alina3 with password='code'
create user alina3 for login alina3

create login shumile5 with password='code'
create user shumile5 for login shumile5

create login munir6 with password='code'
create user munir6 for login munir6

create login nawaz12 with password='code'
create user nawaz12 for login nawaz12

-- login for the employees
create login iftikhar7 with password='code'
create user iftikhar7 for login iftikhar7

create login daud8 with password='code'
create user daud8 for login daud8

create login hassan9 with password='code'
create user hassan9 for login hassan9

create login waqas10 with password='code'
create user waqas10 for login waqas10

create login waheed11 with password='code'
create user waheed11 for login waheed11

create login fazal13 with password='code'
create user fazal13 for login fazal13
```

```sql
--login for the clients

create login faizan14 with password='code'
create user faizan14 for login faizan14

create login johnny15 with password='code'
create user johnny15 for login johnny15

create login farheen16 with password='code'
create user farheen16 for login farheen16

create login kirby20 with password='code'
create user kirby20 for login kirby20

create login ahemd21 with password='code'
create user ahmed21 for login ahemd21

create login rachella22 with password='code'
create user rachella22 for login rachella22

--login for the doctors

create login faisal26 with password='code'
create user faisal26 for login faisal26

create login mohsin27 with password='code'
create user mohsin27 for login mohsin27

create login faizana28 with password='code'
create user faizana28 for login faizana28

create login imea29 with password='code'
create user imea29 for login imea29

create login ghalib30 with password='code'
create user ghalib30 for login ghalib30

create login noror31 with password='code'
create user noror31 for login noror31

--login for the salesmen and saleladies

create login usama17 with password='code'
create user usama17 for login usama17

create login waheed18 with password='code'
create user waheed18 for login waheed18

create login regina19 with password='code'
create user regina19 for login regina19

create login arlie23 with password='code'
create user arlie23 for login arlie23

create login loren24 with password='code'
create user loren24 for login loren24

create login tracy25 with password='code'
```

```sql
create user tracy25 for login tracy25

--login for the Techincal staff is here

create login harry32 with password='code'
create user harry32 for login harry32


create login faisala33 with password='code'
create user faisala33 for login faisala33

create login nazakat34 with password='code'
create user nazakat34 for login nazakat34

create login ayesha35 with password='code'
create user ayesha35 for login ayesha35

create login xavier36 with password='code'
create user xavier36 for login xavier36
```

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52))* - Microsoft SQL Server Management Studio

```
select *from General_info;
-- I have used dintinct here to get unique values
select distinct First_name from General_info;
--I have used ascending and descending here
select* from General_info order by DOB;
select*from General_info order by GID asc;
select*from General_info order by  GID desc;

--create the table for admin use
--I have created table no.1 here
```

	GID	First_name	Last_name	CNIC	DOB	Gender	Marital_Status	Address	Phone_number	job_title
1	1	khurram	shahzad	34202-2412785-9	12-12-2000	M	Unmarried	Kotla arab ali khan	3177647269	Manager
2	2	Robert	kiyoski	32302-2412785-9	11-12-1999	M	Married	America	318494469	Admin
3	3	alina	noor	34202-24755785-9	11-2-1981	F	Unmarried	Gujrat	317383269	Manager
4	4	Ammarah	Khalid	34202-2414785-9	03-3-1999	F	married	Sialkot	3144004825	Manager
5	5	Shumile	Mirza	24202-34785-9	01-1-1985	M	Unmarried	Gujrat	337344884	Manager
6	6	Munir	Butt	14202-241237385-1	02-1-2000	M	married	karnana	238495959	Manager
7	7	iftihar	akhtar	02202-2848485-2	03-9-1960	M	married	malikpur	3184885269	Employee
8	8	daud	kim	23202-234785-9	04-4-2022	M	Unmarried	sailakot	124847269	Employee
9	9	hassan	tahir	32342-2495585-0	05-12-1991	M	married	lahore	319494433	Employee
10	10	waqas	fazal	12202-2414945-7	04-12-2000	M	married	karachi	313437269	Employee
11	11	waheed	aslam	14202-3412784-5	11-2-1972	M	Unmarried	korangi	3173343469	Employee
12	12	nawaz	khan	33202-23343785-6	06-03-1999	M	divorced	peshawar	113849333	Manager

Query executed successfully. DESKTOP-6UPAH3I (12.0 RTM) DESKTOP-6UPAH3I\UsEr (52) dbms_project1 00:00:00 36 row

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52)) - Microsoft SQL Server Management Studio

```
--aggregat functions has been used here
select sum(salary) as all_managers_pay from Manager_table;
select max(salary) as highest_pay from Manager_table;
select min(salary) as minimum_pay from Manager_table;
select avg(salary) as Average_salary from Manager_table;
select count(salary ) as total_managers from Manager_table;

--commands to run the managers code
```

	GID	First_name	Last_name	CNIC	DOB	Gender	Marital_Status	Address	Phone_number	job_title
1	1	khurram	shahzad	34202-2412785-9	12-12-2000	M	Unmarried	Kotla arab ali khan	3177647269	Manager
2	2	Robert	kiyoski	32302-2412785-9	11-12-1999	M	Married	America	318494469	Admin
3	3	alina	noor	34202-24755785-9	11-2-1981	F	Unmarried	Gujrat	317383269	Manager
4	4	Ammarah	Khalid	34202-2414785-9	03-3-1999	F	married	Sialkot	3144004825	Manager

	Admin_id	salary	experinence_before	worked_for	GID
1	11	200000	19	NA	2

	Manager_id	working_in	worked_for	salary	worked_under	GID	Admin_id
1	12	burraq pharmacy	itfaq pharmacy	128494	robert kiyoski	1	11
2	13	itfaq pharmacy	burraq pharmacy	128494	robert kiyoski	4	11
3	14	pure pharmacy	shoot pharmacy	142494	robert kiyoski	3	11
4	15	liaza pharmacy	pure pharmacy	1754...	robert kiyoski	6	11
5	16	horbon pharma...	shareef pharm...	1434...	robert kiyoski	12	11
6	17	Noor pharmacy	kareem pharm...	12844	robert kiyoski	5	11

Query executed successfully. DESKTOP-6UPAH3I (12.0 RTM) DESKTOP-6UPAH3I\UsEr (52) dbms_project1 00:00:00 1 rows

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52)) - Microsoft SQL Server Management Studio

```
173 ------------------------------------------------------------------------------
174                                                                            --
175 select*from General_info;                                                  --
176 select*from Manager_table;                                                 --
177 select*from Employess;                                                     --
178                                                                            --
179 ------------------------------------------------------------------------
180 -- I have created the table no.4 here
181 --I have created this table for the client data
182
```

Results | Messages

	Employee_id	working_in	worked_for	salary	experience	working_under	Manager_id	GID
1	21	burraq pharmacy	itfaq pharmacy	12393	3	Ammarah Khalid	13	7
2	22	itfaq pharmacy	burraq pharmacy	12373	4	Khurram Shahzad	12	8
3	23	pure pharmacy	shoot pharmacy	127464	12	alina noor	14	9
4	24	liaza pharmacy	pure pharmacy	122823	4	Shumile mirza	17	10
5	25	horbon pharmacy	shareef pharmacy	148553	8	munir butt	15	11
6	26	noor pharmacy	kareem pharmacy	200023	10	nawaz khan	16	13

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52)) - Microsoft SQL Server Management Studio

```
208                                                                            --
209 select *from client_table;                                                 --
210 select*from General_info;
211 select General_info.First_name,General_info.Last_name,
212 client_table.ordered_medicine,client_table.order_price,client_table.add_to_cart,
213 client_table.total_buying,client_table.buying_pharmacy,
214 client_table.linked_doctor,client_table.disease_to_patient
215 from General_info
216 inner join client_table on General_info.GID=client_table.GID;
217                                                                            --
```

Results | Messages

	GID	First_name	Last_name	CNIC	DOB	Gender	Marital_Status	Address	Phone_number	job_title
1	1	khurram	shahzad	34202-2412785-9	12-12-2000	M	Unmarried	Kotla arab ali khan	3177647269	Manager
2	2	Robert	kiyoski	32302-2412785-9	11-12-1999	M	Married	America	318494469	Admin
3	3	alina	noor	34202-24755785-9	11-2-1981	F	Unmarried	Gujrat	317383269	Manager
4	4	Ammarah	Khalid	34202-2414785-9	03-3-1999	F	married	Sialkot	3144004825	Manager
5	5	Shumile	Mirza	24202-34785-9	01-1-1985	M	Unmarried	Gujrat	337344884	Manager
6	6	Munir	Butt	14202-241237385-1	02-1-2000	M	married	karnana	238495959	Manager
7	7	iftihar	akhtar	02202-2848485-2	03-9-1960	M	married	malikpur	3184885269	Employee
8	8	daud	kim	23202-234785-9	04-4-2022	M	Unmarried	sailakot	124847269	Employee

	First_name	Last_name	ordered_medicine	order_price	add_to_cart	total_buying	buying_pharmacy	linked_doctor	disease_to_patient
1	johnny	james	panadol	320	lipiget	1000	burraq pharmacy	faisal qureshi	Headache
2	farheen	nasira	lipiget	340	panadol	2000	itfaq pharmacy	ghalib khan	blood sugar
3	faizan	khakid	asphil	360	lipirex	3000	noor pharmacy	noror badavia	fever
4	kirby	richard	getryl	380	asphil	4000	liaza pharmacy	faizan syeda	TB
5	ahmed	raza	calcuim-c	400	getryl	5000	pure pharmacy	mohsin habib	diearah
6	rachella	robinson	lipirex	420	calcuim-C	6000	horbon pharmacy	imea falata	heart problem

Query executed successfully. DESKTOP-6UPAH3I (12.0 RTM) DESKTOP-6UPAH3I\UsEr (52) dbms_project1 00:00:00 42 rows

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52)) - Microsoft SQL Server Management Studio

```sql
select*from Doctors_table;
select*from client_table;
select*from General_info;
```

	client_id	ordered_medicine	order_price	add_to_cart	total_buying	buying_pharmacy	linked_doctor	disease_to_patient	GID
1	31	panadol	320	lipiget	1000	burraq pharmacy	faisal qureshi	Headache	15
2	32	lipiget	340	panadol	2000	itfaq pharmacy	ghalib khan	blood sugar	16
3	33	asphil	360	lipirex	3000	noor pharmacy	noror badavia	fever	14
4	34	getryl	380	asphil	4000	liaza pharmacy	faizan syeda	TB	20

	GID	First_name	Last_name	CNIC	DOB	Gender	Marital_Status	Address	Phone_number	job_title
1	1	khurram	shahzad	34202-2412785-9	12-12-2000	M	Unmarried	Kotla arab ali khan	3177647269	Manager
2	2	Robert	kiyoski	32302-2412785-9	11-12-1999	M	Married	America	318494469	Admin
3	3	alina	noor	34202-2475578...	11-2-1981	F	Unmarried	Gujrat	317383269	Manager
4	4	Ammarah	Khalid	34202-2414785-9	03-3-1999	F	married	Sialkot	3144004825	Manager
5	5	Shumile	Mirza	24202-34785-9	01-1-1985	M	Unmarried	Gujrat	337344884	Manager
6	6	Munir	Butt	14202-2412373...	02-1-2000	M	married	kamana	238495959	Manager
7	7	iftihar	akhtar	02202-2848485-2	03-9-1960	M	married	malikpur	3184885269	Employ...
8	8	daud	kim	23202-234785-9	04-4-2022	M	Unmarried	sailakot	124847269	Employ...

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52)) - Microsoft SQL Server Management Studio

```sql
select General_info.First_name,General_info.Last_name,
General_info.Gender,Salesmen_table.Salesmen_id,Salesmen_table.stock_sold,Salesmen_table.worked_for,
Salesmen_table.earned_commission,Salesmen_table.withdrew_commission,
Salesmen_table.worked_years,Salesmen_table.working_under,
Salesmen_table.Manager_id
from General_info inner join
Salesmen_table on General_info.GID=Salesmen_table.Manager_id;
```

	First_name	Last_name	Gender	Salesmen_id	stock_sold	worked_for	earned_commission	withdrew_commission	worked_years	working_under	Manager_id
1	nawaz	khan	M	51	panadol	itfaq pharmacy	5000	1000	2	Khurram Shahzad	12
2	fazal	kakad	M	52	lipiget	burraq pharma	6000	2000	3	Ammarah Khalid	13
3	faizan	khakid	M	53	lipirex	shoot pharmacy	7000	3000	4	Alina Noor	14
4	johnny	james	M	54	getryl	pure pharmacy	8000	4000	5	Munir Butt	15
5	johnny	james	M	55	asphil	pure pharmacy	9000	5000	6	Munir Butt	15
6	farheen	nasira	F	56	Calcuim-C	shareef pharmacy	10000	6000	7	Nawaz Khan	16

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52)) - Microsoft SQL Server Management Studio

```
27,500,'nupuria pharma','1-7-2023',24,54)
insert into inventory_table values(66,'asphil',110,'pure pharmacy','hassan tahir','tracy wolf',
28,600,'atgate pharma','12-2-2020',23,53)

--------------------------------------------------------------------------------

select*from inventory_table;

--------------------------------------------------------------------------------
```

	inventory_id	Medicine_name	medicine_price	destination_pharmacy	sold_by_employee	salesmen_connected	stock_sold_out	stock_available	source_of_medicine	expiry_date	Employee_
1	61	panadol	60	burraq pharmacy	johnny james	usama aslam	23	100	al-shareef pharma	31-12-2023	21
2	62	lipiget	70	itfaq pharmacy	daud kim	waheed fasani	24	200	careem pharma	31-1-2024	22
3	63	lipirex	80	horbon pharmacy	waheed aslam	tracy wolf	25	300	safina pharma	12-12-2026	25
4	64	getryl	90	noor pharmacy	fazal kakad	NA	26	400	barqat pharma	9-05-2023	26
5	65	calcium-c	100	liaza pharmacy	waqas fazal	loren byrd	27	500	nupuria pharma	1-7-2023	24
6	66	asphil	110	pure pharmacy	hassan tahir	tracy wolf	28	600	atgate pharma	12-2-2020	23

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52))* - Microsoft SQL Server Management Studio

```
insert into tech_staff_table values (71,'burraq pharmacy','khurram shahzad',15000,32 ,1)
insert into tech_staff_table values (72,'itfaq pharmacy','Ammarah khalid',20000,33 ,4)
insert into tech_staff_table values (73,'pure pharmacy','Alina noor ',25000, 34,3)
insert into tech_staff_table values (74,'liaza pharmacy','Munir Butt',30000,35 ,6)
insert into tech_staff_table values (75,'noor pharmacy','Shumile Mirza',35000,36 ,5)
select*from tech_staff_table
```

	tech_staff_id	working_in	working_under	salary	GID	Manger_id
1	71	burraq pharmacy	khurram shahzad	15000	32	1
2	72	itfaq pharmacy	Ammarah khalid	20000	33	4
3	73	pure pharmacy	Alina noor	25000	34	3
4	74	liaza pharmacy	Munir Butt	30000	35	6
5	75	noor pharmacy	Shumile Mirza	35000	36	5

projectfile.sql - DESKTOP-6UPAH3I.dbms_project1 (DESKTOP-6UPAH3I\UsEr (52))* - Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Help

New Query | salesmen | dbms_project1 | Execute | Debug

Object Explorer

Connect

- Users
  - ahmed21
  - alina3
  - ammarah4
  - arlie23
  - ayesha35
  - daud8
  - dbo
  - faisal26
  - faisala33
  - faizan14
  - faizana28
  - farheen16
  - fazal13
  - ghalib30
  - guest
  - harry32
  - hassan9
  - iftikhar7
  - imea29
  - INFORMATION_SCHEMA
  - johnny15
  - khurram
  - khurram1
  - kirby20
  - loren24

Updated.sql - DESK...6UPAH3I\UsEr (54)) | projectfile.sql - D...P-6UPAH3I\UsEr (52))*

```
insert into login values('waqas10','code',10)
insert into login values('waheed11','code',11)
insert into login values('nawaz12','code',12)
insert into login values('fazal13','code',13)
insert into login values('faizan14','code',14)
insert into login values('johnny15','code',15)
insert into login values('farheen16','code',16)
insert into login values('usama17','code',17)
insert into login values('waheed18','code',18)
insert into login values('regina19','code',19)
insert into login values('kirby20','code',20)
insert into login values('ahmed21','code',21)
insert into login values('rachella22','code',22)
insert into login values('arlie23','code',23)
insert into login values('loren24','code',24)
insert into login values('tracy25','code',25)
insert into login values('faisal26','code',26)
insert into login values('mohsin27','code',27)
insert into login values('faizana28','code',28)
insert into login values('imea29','code',29)
insert into login values('ghalib30','code',30)
insert into login values('noror31','code',31)
insert into login values('harry32','code',32)
```

100 %

Messages

```
Msg 156, Level 15, State 1, Procedure all_personal_data, Line 380
Incorrect syntax near the keyword 'select'.
```

100 %