PROJECT

MEDICAL STORE MANAGEMENT

Introduction of the Project Medical Shop Management System

The "Medical Shop Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner. The application is reduced as much as possible to avoid errors while entering the data .It also provides error message while entering invalid data No formal knowledge is needed for the user to use this system

NARRATIVE OF PROJECT:

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

Objective of Project on Medical Shop Management System:

The main objective of the Project on Medical Shop Management System is to manage the details of Medical Shop, Medicines, Stocks, Company, Inventory It manages all the information about Medical Shop, Sells, Inventory, Medical Shop.

The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Medical Shop, Medicines, Sells, Stocks. It tracks all the details about the Stocks, Company, Inventory.

Functionalities provided by Medical Shop Management System are as follows:

- Provides the searching facilities based on various factors. Such as Medical Shop, Stocks, Company, Inventory
- Medical Shop Management System also manage the Sells details online for Company details, Inventory details, Medical Shop.
- It tracks all the information of Medicines, Sells, Company etc
- Manage the information of Medicine Shows the information and description of the Medical Shop, Stocks
- To increase efficiency of managing the Medical Shop, Medicines
- It deals with monitoring the information and transactions of Company.
- Manage the information of Medical Shop

Scope of the project Medical Shop Management System

It may help collecting perfect management in detail. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Medical Shop Management System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

Our project aims at

- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time
- To assist the pharmacist in capturing the effort spent on their respective working areas.

- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfies the user requirement
- Be easy to understand by the user and operator
- Be easy to operate
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget.

STAKEHOLDERS:

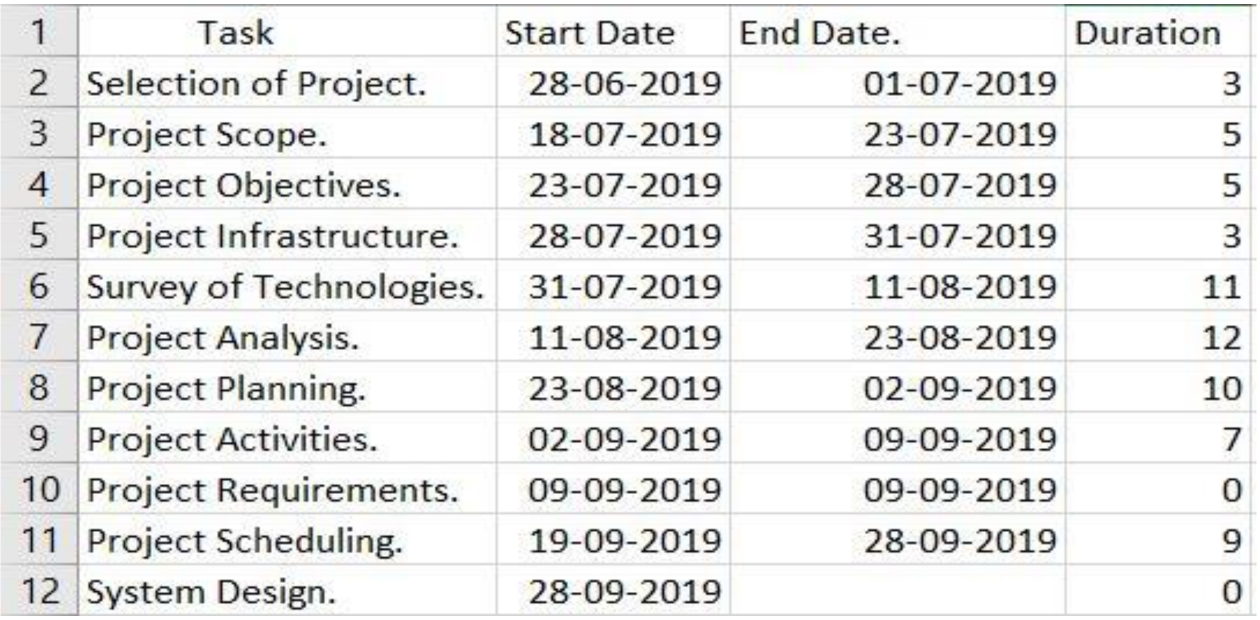
- ➢ Admin
- ➢ Manager
- ➢ Pharmacist
- ➢ Cashier

Stakeholder’s Role:

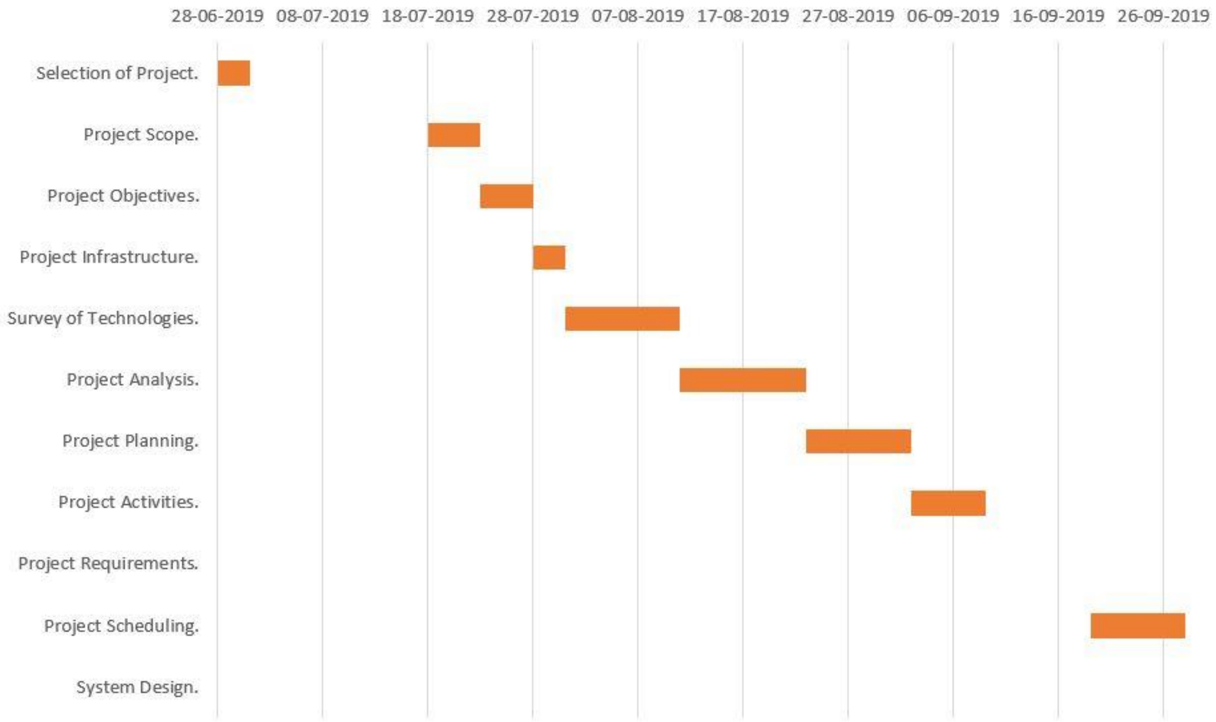
- ❑ **Admin:** Admin looks after the store adds and removes Cashier to the website. Adds and removes pharmacist to the website. Manages all the information about the pharmacist and Cashier.
- ❑ **Pharmacist:** He can be able to add the new medicines information. And he can view the expired medicines information and he can be able to update the medicines status. He behaves like the manufacturer of medicines.
- ❑ **Cashier:** Process the payments.
- ❑ **Manager:** Manages all the pharmacist and cashier details.

Gantt chart:

- Gantt chart is a graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion.
- Different colors, shades or shapes can be used to highlight each kind of task.
- So, this Project Plan begins from the third week of June and lasting till the last week September.
- It comprises of lots of hard work which will enumerated step by step in the following Gantt chart: `



1	Task	Start Date	End Date.	Duration
2	Selection of Project.	28-06-2019	01-07-2019	3
3	Project Scope.	18-07-2019	23-07-2019	5
4	Project Objectives.	23-07-2019	28-07-2019	5
5	Project Infrastructure.	28-07-2019	31-07-2019	3
6	Survey of Technologies.	31-07-2019	11-08-2019	11
7	Project Analysis.	11-08-2019	23-08-2019	12
8	Project Planning.	23-08-2019	02-09-2019	10
9	Project Activities.	02-09-2019	09-09-2019	7
10	Project Requirements.	09-09-2019	09-09-2019	0
11	Project Scheduling.	19-09-2019	28-09-2019	9
12	System Design.	28-09-2019		0


28-06-2019
08-07-2019
18-07-2019
28-07-2019
07-08-2019
17-08-2019
27-08-2019
06-09-2019
16-09-2019
26-09-2019
Selection of Project.
Project Scope.
Project Objectives.
Project Infrastructure.
Survey of Technologies.
Project Analysis.
Project Planning.
Project Activities.
Project Requirements.
Project Scheduling.
System Design.

Software and Hardware specification:

Hardware Specification:

- **Server Side:**
  - Processor: 2.0 GHZ
  - RAM: 2 GB
  - Hard Disk: 30 GB free space

- **Client Side:**
  - Processor: 1.0 GHZ
  - RAM: 512 MB
  - Hard Disk: 2 GB free space

Software Specification:

- **Front End:** PHP

- **Back End:** SQL Server 2008

Technologies Description

PHP:

PHP (recursive acronym for *PHP: Hypertext Preprocessor*) is a widely used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

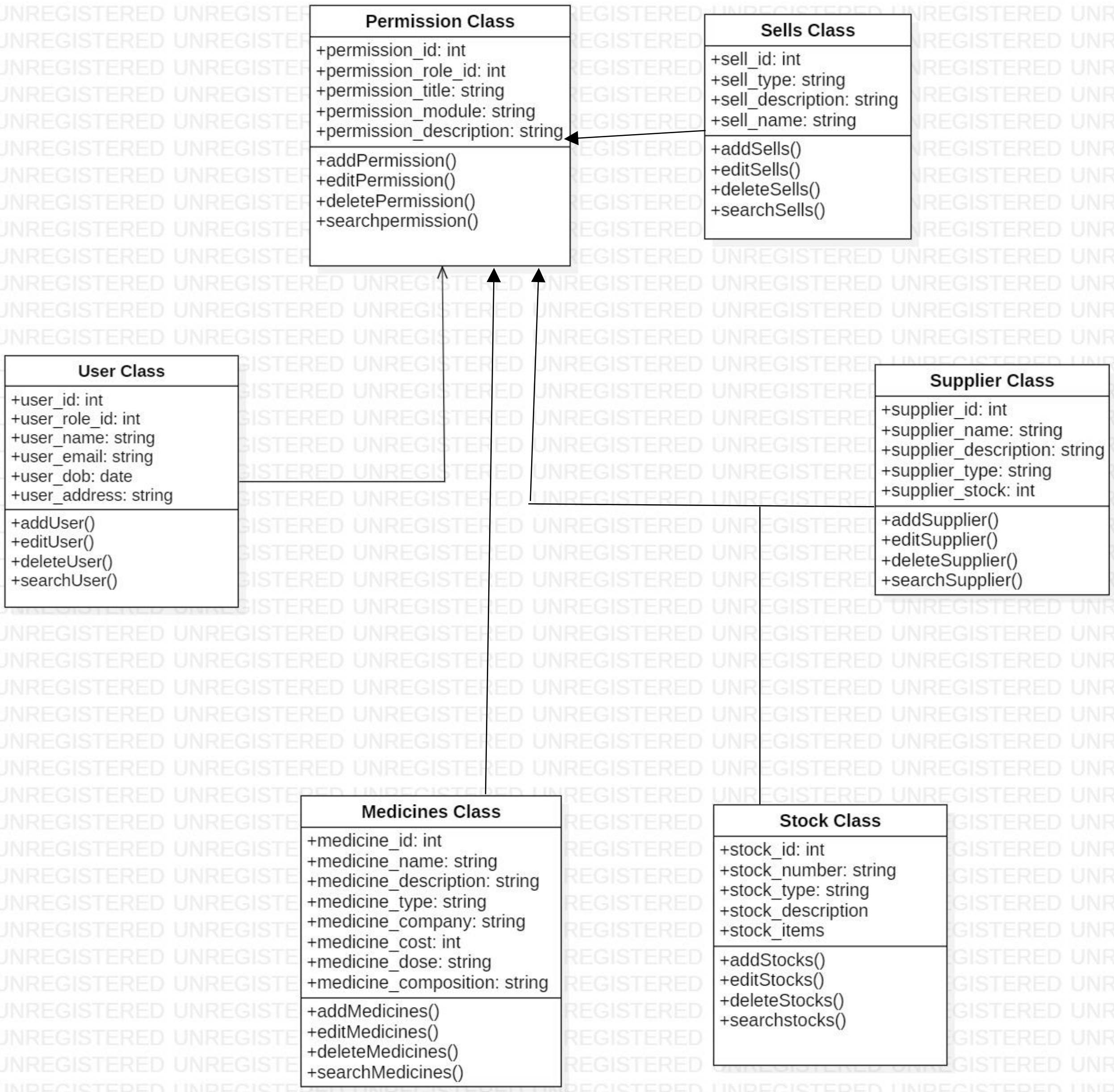
Fast & Smart Code Editing

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code — which may be any type of data, such as generated HTML or binary image data — would form the whole or part of a HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response.

UML Diagrams

Class diagram:

- A class diagram in the Unified Modified Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
- In the diagram, classes are represented with boxes that contain three compartments:
- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned, and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned, and the first letter is lowercase.

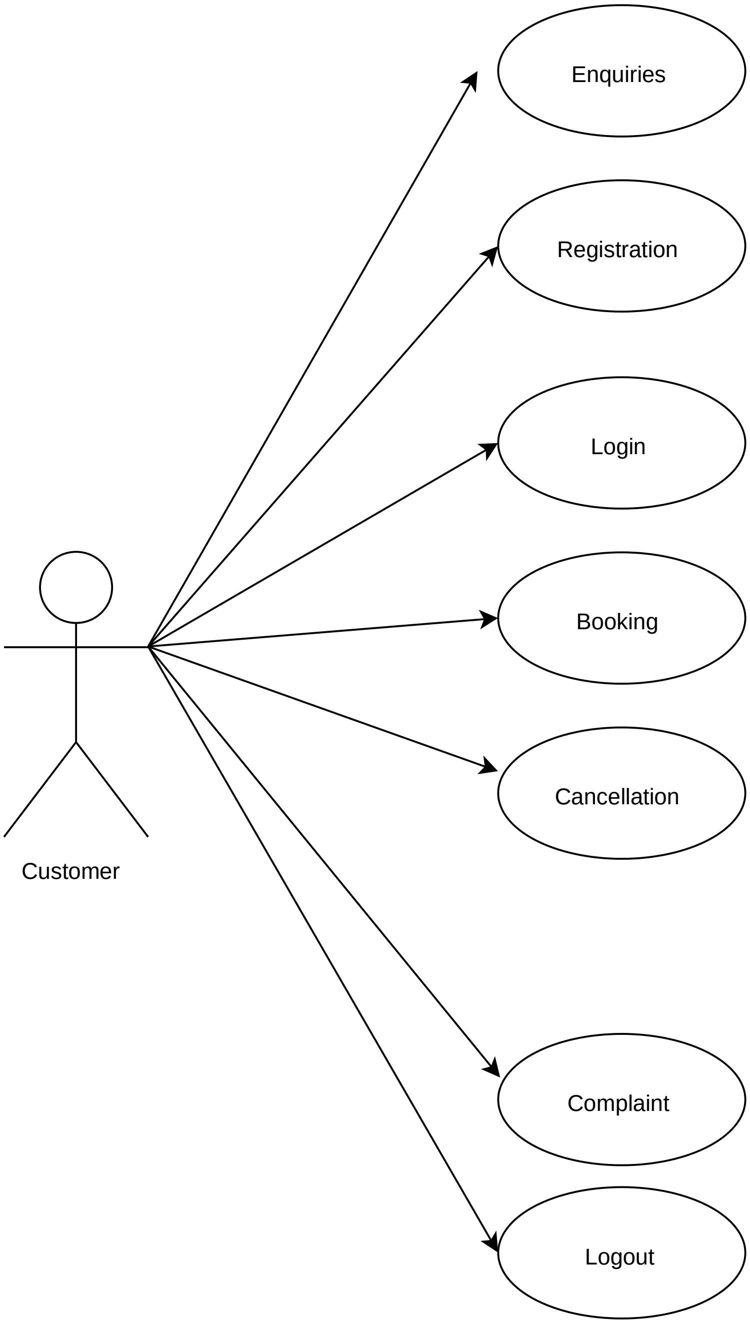

Permission Class
+permission_id: int
+permission_role_id: int
+permission_title: string
+permission_module: string
+permission_description: string
+addPermission()
+editPermission()
+deletePermission()
+searchpermission()
Sells Class
+sell_id: int
+sell_type: string
+sell_description: string
+sell_name: string
+addSells()
+editSells()
+deleteSells()
+searchSells()
User Class
+user_id: int
+user_role_id: int
+user_name: string
+user_email: string
+user_dob: date
+user_address: string
+addUser()
+editUser()
+deleteUser()
+searchUser()
Supplier Class
+supplier_id: int
+supplier_name: string
+supplier_description: string
+supplier_type: string
+supplier_stock: int
+addSupplier()
+editSupplier()
+deleteSupplier()
+searchSupplier()
Medicines Class
+medicine_id: int
+medicine_name: string
+medicine_description: string
+medicine_type: string
+medicine_company: string
+medicine_cost: int
+medicine_dose: string
+medicine_composition: string
+addMedicines()
+editMedicines()
+deleteMedicines()
+searchMedicines()
Stock Class
+stock_id: int
+stock_number: string
+stock_type: string
+stock_description
+stock_items
+addStocks()
+editStocks()
+deleteStocks()
+searchstocks()

Use case:

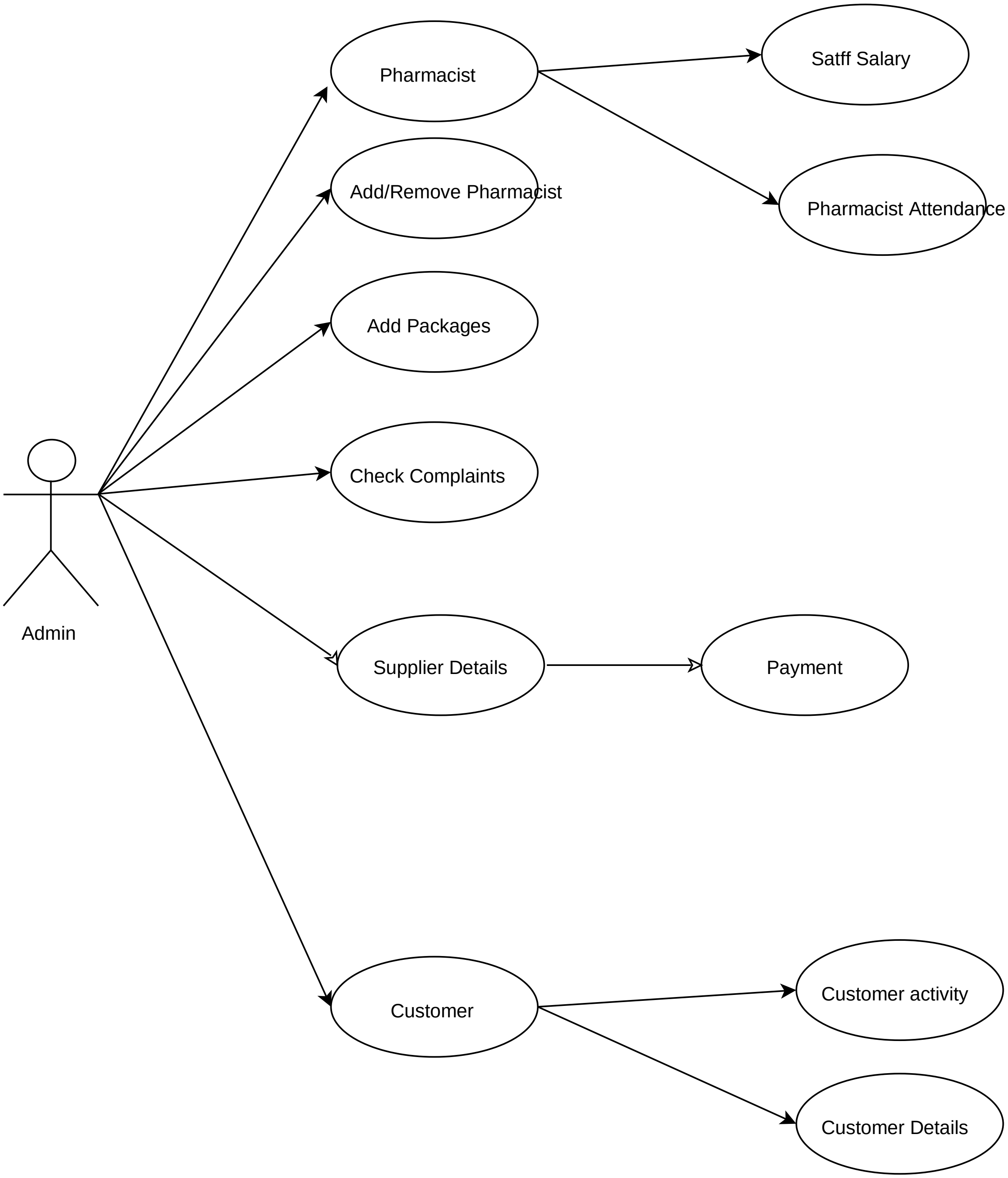
- A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.
- A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.
- The use cases are represented by either circles or ellipses.
- An effective use case diagram can help your team discuss and represent:

  - Scenarios in which your system or application interacts with people, organizations, or external systems
  - Goals that your system or application helps those entities (known as actors) achieve
  - The scope of your system.

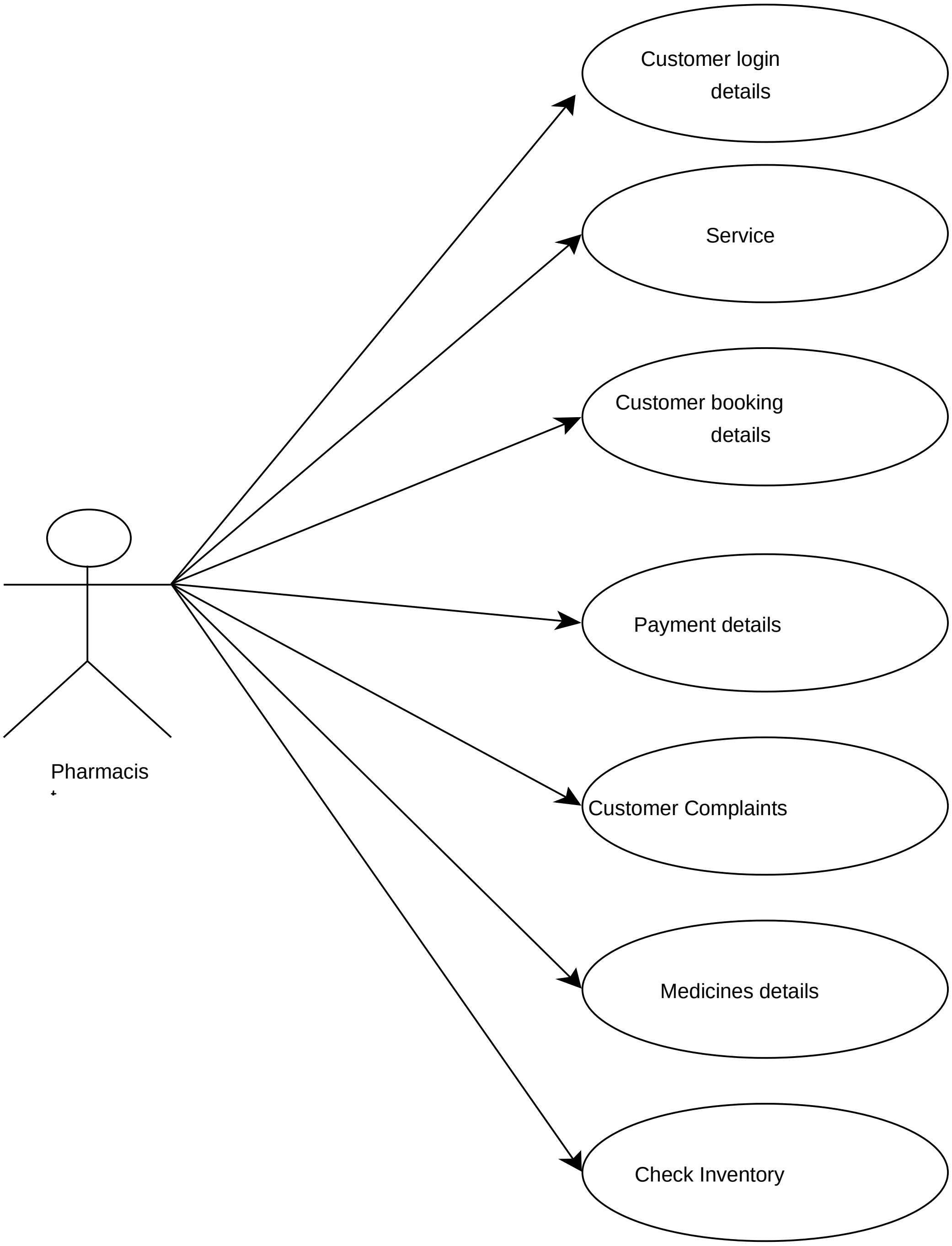
USE CASE-CASHIER



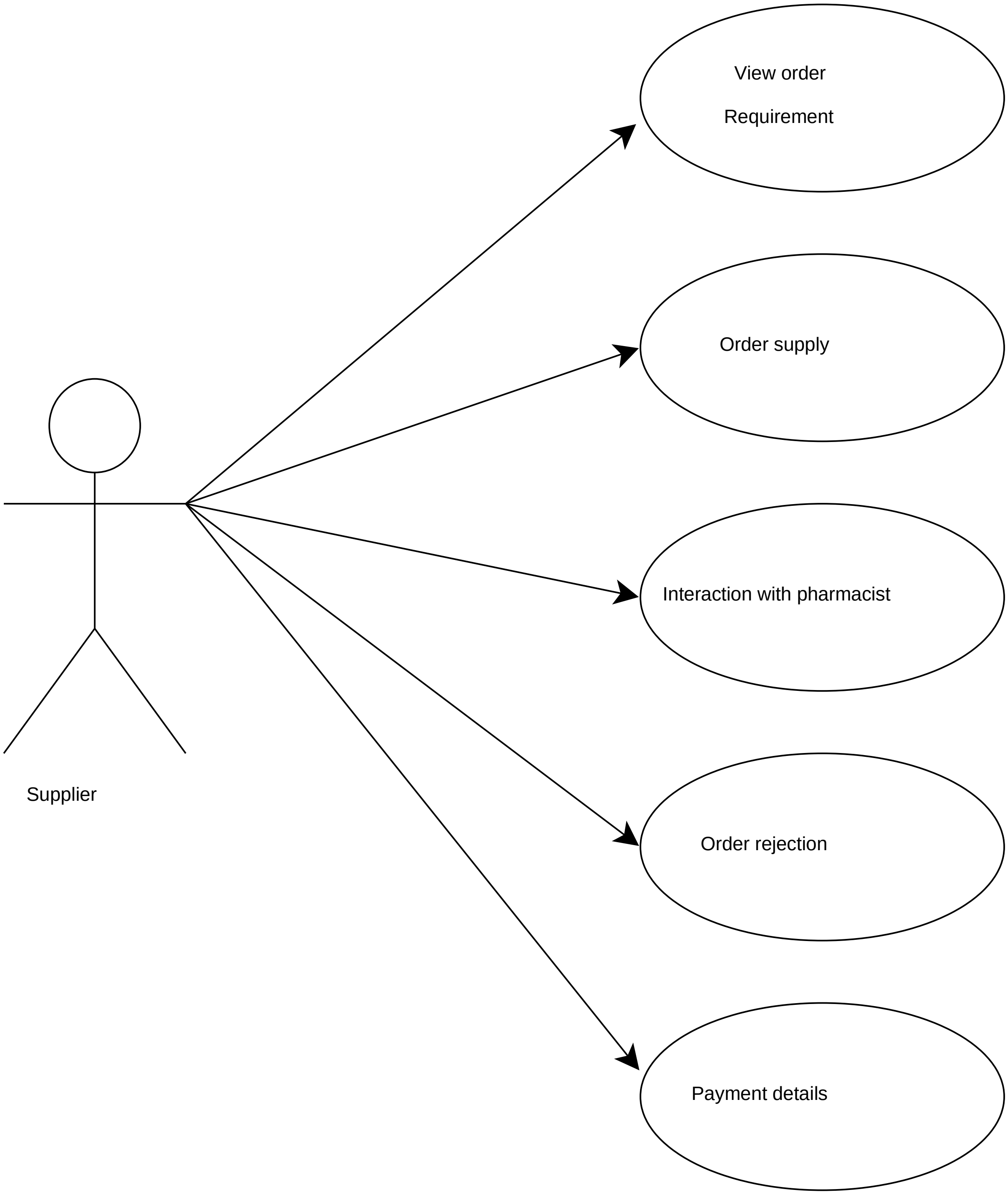
USE CASE-ADMIN



USE CASE-PHARMACIST

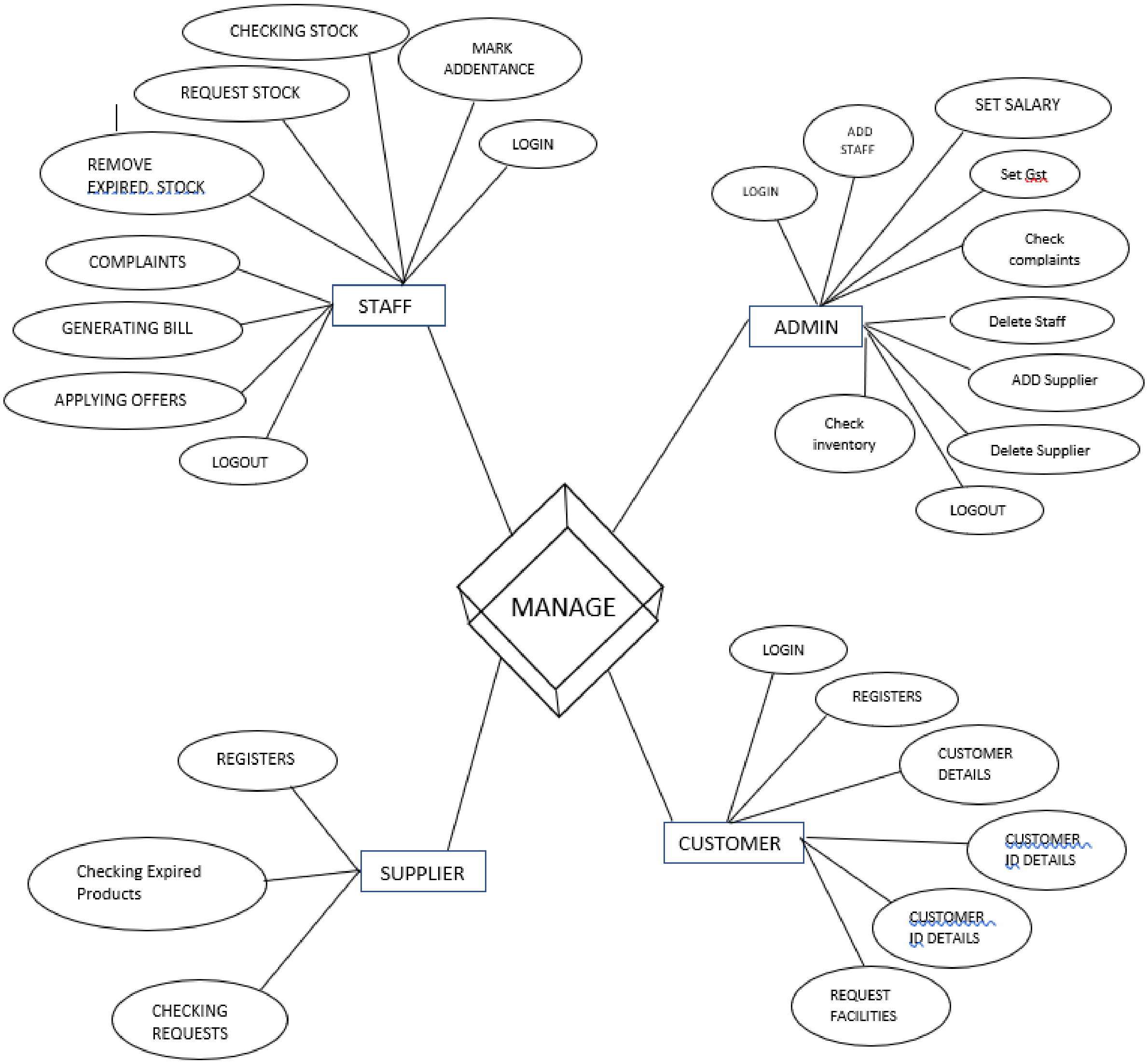


USE CASE-SUPPLIER



Entity Relationship Diagram (ER diagram):

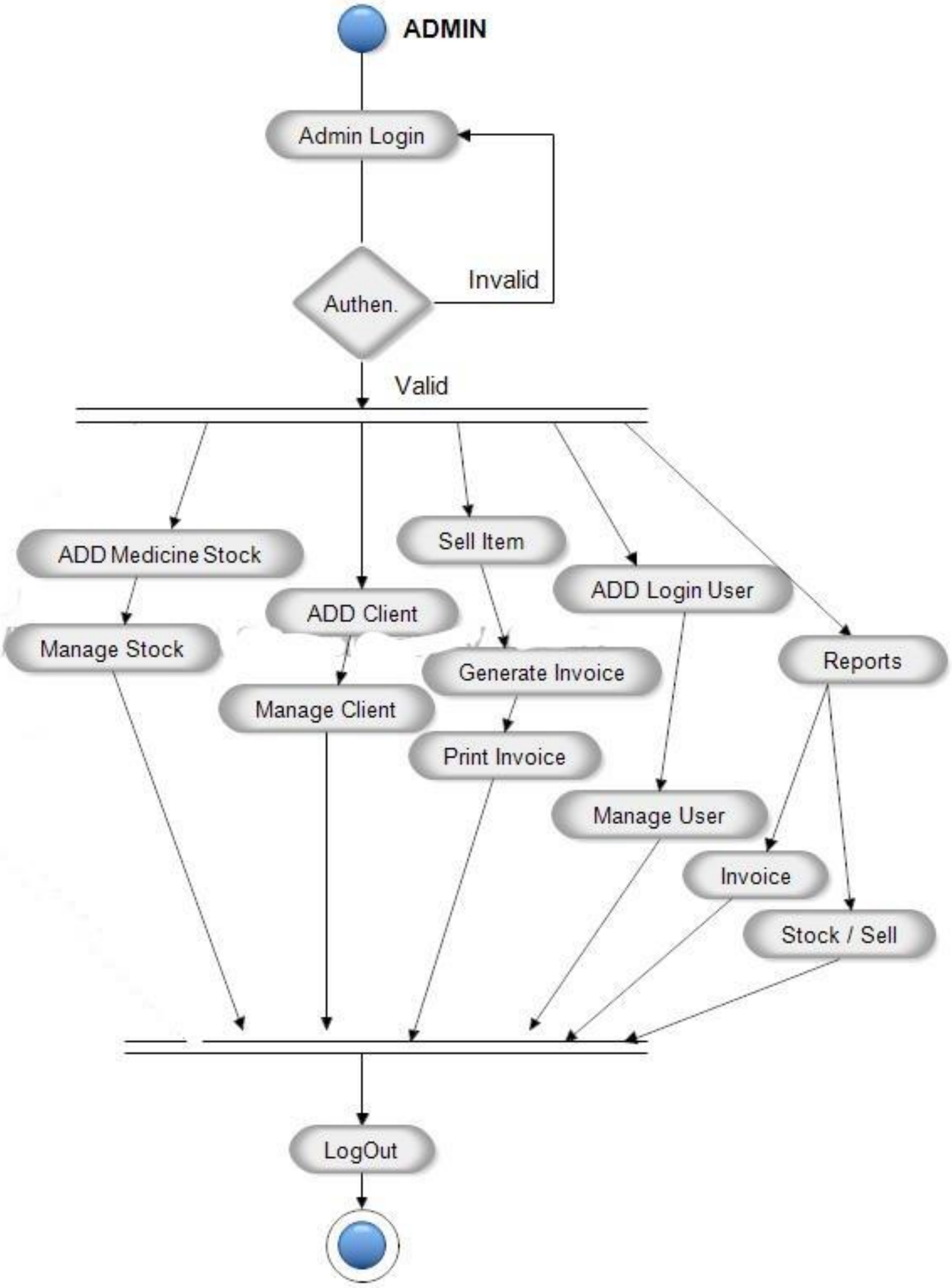
- An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database.
- An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.



Activity Diagram:

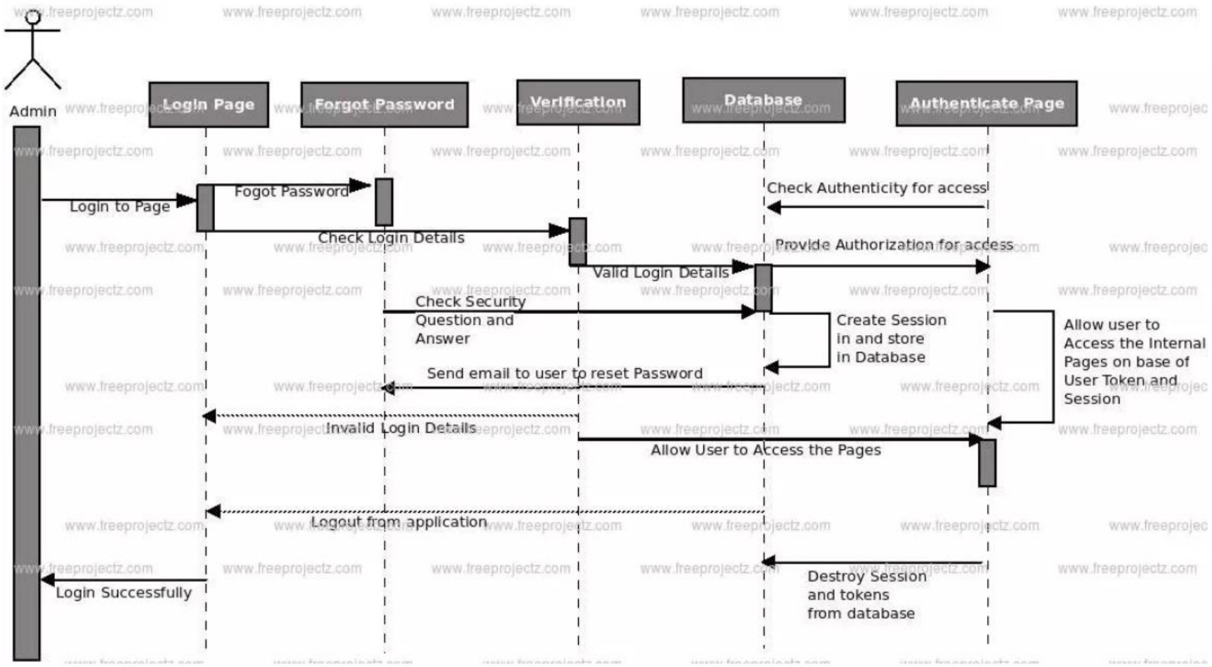
- Activity design is often classified as a computational approach relying upon a set of instructions that, when used in a particular sequence, are the generators of form.
- Software Procedural Design (SPD) converts and translates structural elements into procedural explanations. SPD starts straight after data design and architectural design.
- This has now been mostly abandoned mostly due to the rise in preference of Object-Oriented Programming and design patterns.

Activity Diagram Medical Store System



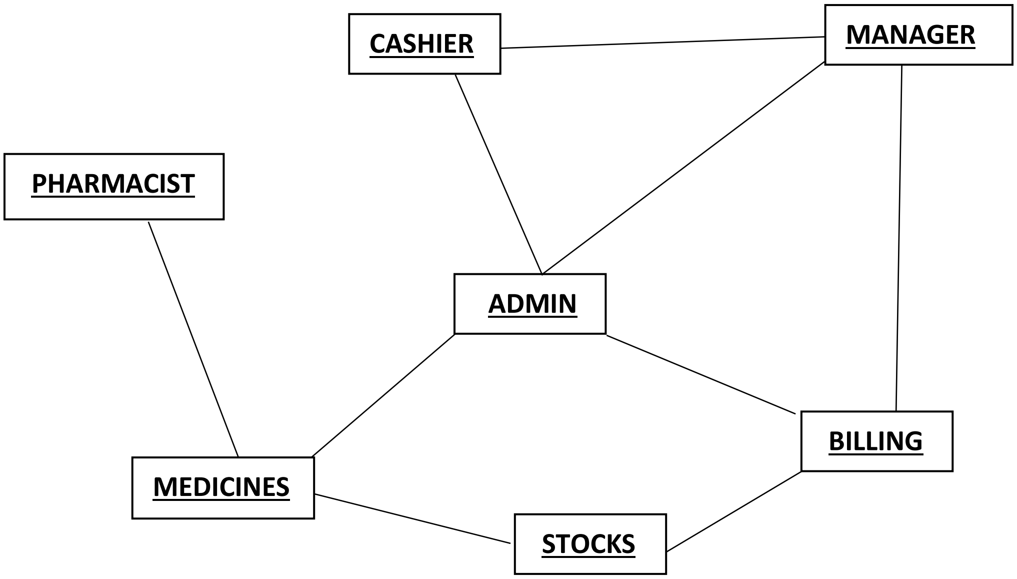
Sequence Diagram

- Sequence diagrams are used in the analysis and design phases.
- Sequence diagrams are often used to depict the chronologically-
- structured event flow through a use case.
- By creating a sequence diagram, the objects that participate in the use case are identified.
- Additionally, pieces of the use case behavior are assigned to objects in the form of services.
- The process of creating a sequence diagram often results in the refinement of the use case, potentially identifying missing but desired behaviors.
- Sequence diagrams represent a system behavior based upon the needed interactions among a set of objects in terms of the messages that exchange among them to produce the desired result.
- Sequence diagrams highlight the sequence of messages through time.
- However, they do not show how objects are linked and may send messages to
- each other.



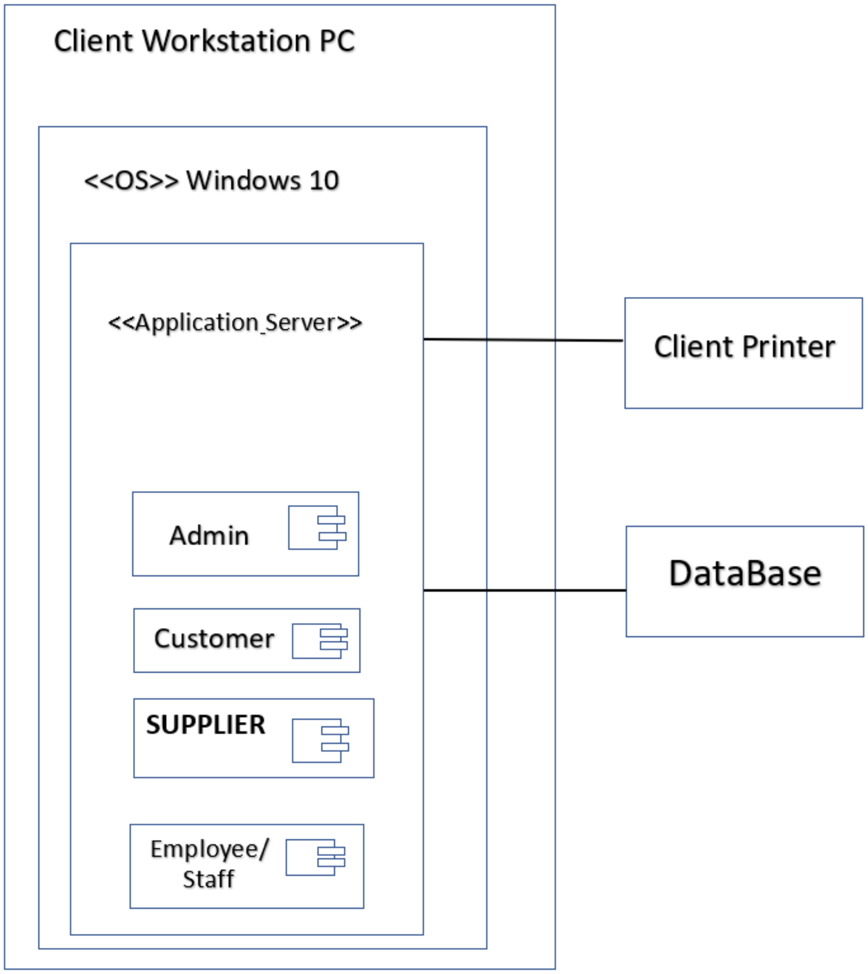
Object Diagram

- Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagram.
- Object diagram represent an instance of a class diagram.
- The basic concepts are similar for class diagrams and object diagrams.
- Object diagrams also represent the static view of a system, but the static view is a snapshot of the system at a particular moment.



Deployment Diagram

- Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets.
- Artifacts represent concrete elements in the physical world that are the result of a development process.
- Examples of artifacts are executable files, libraries, archives, database schemas, configuration files, etc.
- Deployment target is usually represented by a node which is either hardware device or some software execution environment.



EVENT TABLE:

SR No.	EVENT	TRIGGER	SOURCE	ACTIVITY	RESPONSE	DESTINATION
1.	Login	Request of Login	Admin	Login of Admin	Logged in Successfully	Admin
2.	Dashboard	View Dashboard	Admin	Viewing DashBoard	View	Admin
3.	Enter new Pharmacist	Add pharmacist	Admin	Create new pharmacist record	New record created	Admin
4.	Enter new Cashier	Add Cashier	Admin	Create new Cashier record	New record created	Admin
5.	Enter new Manager	Add Manager	Admin	Create new Manager record	New record created	Admin
6.	Deleting Existing Pharmacists	Delete Pharmacists	Admin	Delete Specific Pharmacist	Record Deleted	Admin Pharmacist
7.	Deleting Existing Cashier	Delete Cashier	Admin	Delete Specific Cashier	Record Deleted	Admin Cashier

8.	Deleting Existing Manager	Delete Manager	Admin	Delete Specific Manager	Record Deleted	Admin Manager
9.	Updating Existing Pharmacist	Update Pharmacist	Admin	Update Specific Pharmacist	Record Updated	Admin Pharmacist
9.	Updating Existing Cashier	Update Cashier	Admin	Update Specific Cashier	Record Updated	Admin Cashier
10.	Updating Existing Manager	Update Manager	Admin	Update Specific Manager	Record Updated	Admin Manager
11.	Log-Out	Admin Log Out	Admin	Log Out of Admin	Logged Out Successfully	Admin
12.	Login	Request of Login	Pharmacist	Login of Pharmacist	Logged in Successfully	Pharmacist
13.	Dashboard	View Dashboard	Pharmacist	Viewing DashBoard	View	Pharmacist
14.	Prescription	View Prescription	Pharmacist	View Prescription	View	Pharmacist

15.	New Prescription	Create New Prescription	Pharmacist	Creating new prescription	Created successfully	Pharmacist
16.	Delete Prescription	Delete Prescription	Pharmacist	Deleting prescription	Deleted successfully	Pharmacist
17.	Stock	View Stock	Pharmacist	View Stock	View	Pharmacist
18.	New Stock	Add New Medicine	Pharmacist	Adding new medicine	Added successfully	Pharmacist
19.	Delete Stock	Delete Stock	Pharmacist	Deleting Stock	Deleted successfully	Pharmacist
20.	Logout	Request of Logout	Pharmacist	Logging out	Logged out successfully	Pharmacist

21.	Login	Request of Login	Cashier	Login of Cashier	Logged in Successfully	Cashier
22.	Dashboard	View Dashboard	Cashier	Viewing DashBoard	View	Cashier
23.	Manage Payments	Add Payment details	Cashier	Adding Payment details	Added successfully	Cashier
24.	Logout	Request of Logout	Cashier	Logging out	Logged out successfully	Cashier
25.	Login	Request of Login	Manager	Login of Manager	Logged in Successfully	Manager
26.	Dashboard	View Dashboard	Manager	Viewing DashBoard	View	Manager
27.	View Users	View users	Manager	Viewing all users	View	Manager

27.	View Prescription	View Prescription	Manager	Viewing all Prescription	View	Manager
28.	View Stocks	View stocks	Manager	Viewing all stocks	View	Manager
29.	Add Medicines	Request of adding	Manager	Adding Medicines	Added successfully	Manager
30.	Deleting Medicines	Request of Deleting	Manager	Deleting Medicines	Deleted successfully	Manager
31.	Logout	Request of Logout	Manager	Logging out	Logged out successfully	Manager

CONCLUSION:

Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a framework that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points...

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project.
- The description of Purpose, Scope, and applicability
- We define the problem on which we are working in the project
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts