**Online Grocery System**

**A PROJECT REPORT**

**Submitted By**

**Aaditya Singh**

**2000290140001**

**Jai pratap singh**

**2000290140053**

**Kshitiz Pandey**

**2000290140060**

**Omveer Singh**

**2000290140082**

**Submitted in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATIONS**

**Under the Supervision of**

**Ms. Shalika Arora**

### Assistant Professor



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(Jan 2022)**

**CERTIFICATE**

Certified that **Aaditya Singh (2000290140001), Jai pratap singh (2000290140053), Kshitiz Pandey (2000290140060), Omveer Singh (2000290140082)**

have carried out the project work having “**Online Grocery System**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date: 15/01/2022**

**Aaditya Singh**

**2000290140001**

**Jai pratap singh**

**2000290140053**

**Kshitiz Pandey**

**2000290140060**

**Omveer Singh**

**2000290140082**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 15/01/2022

**Ms. Shalika Arora**

**Assistant Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Ajay Shrivastava**

**Head, Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| [1.](#_bookmark0) | [Introduction](#_bookmark0) | [5](#_bookmark0) |
| [1.1 Purpose](#_bookmark1) | | [5](#_bookmark1) |
| [1.2 Document Conventions](#_bookmark2) | | [5](#_bookmark2) |
| [1.3 Intended Audience and Reading Suggestions](#_bookmark3) | | [5](#_bookmark3) |
| [1.4 Product Scope](#_bookmark4) | | [5](#_bookmark4) |
| [1.5 References](#_bookmark5) | | [6](#_bookmark5) |
| [2.](#_bookmark6) | [Overall Description](#_bookmark6) | [6](#_bookmark6) |
| [2.1 Product Perspective](#_bookmark7) | | [6](#_bookmark7) |
| [2.2 Product Functions](#_bookmark8) | | [6](#_bookmark8) |
| [2.3 User Classes and Characteristics](#_bookmark9) | | [7](#_bookmark9) |
| [2.4 Operating Environment](#_bookmark10) | | [7](#_bookmark10) |
| [2.5 Design and Implementation Constraints](#_bookmark11) | | [7](#_bookmark11) |
| [2.6 User Documentation](#_bookmark12) | | [7](#_bookmark12) |
| [2.7 Assumptions and Dependencies](#_bookmark13) | | [7](#_bookmark13) |
| [3.](#_bookmark14) | [External Interface Requirements](#_bookmark14) | [7](#_bookmark14) |
| [3.1 User Interfaces](#_bookmark15) | | [7](#_bookmark15) |
| [3.2 Hardware Interfaces](#_bookmark16) | | [8](#_bookmark16) |
| [3.3 Software Interfaces](#_bookmark17) | | [8](#_bookmark17) |
| [3.4 Communications Interfaces](#_bookmark18) | | [8](#_bookmark18) |
| [4.](#_bookmark19) | [System Features](#_bookmark19) | [8](#_bookmark19) |
| [4.1 Login/Signup Page](#_bookmark20) | | [8](#_bookmark20) |
| [4.2 View Course](#_bookmark21) | | [9](#_bookmark21) |
| [4.3 View User Profile](#_bookmark22) | | [9](#_bookmark22) |
| [4.4 Logout Page](#_bookmark23) | | [9](#_bookmark23) |
| [5.](#_bookmark24) | [Other Nonfunctional Requirements](#_bookmark24) | [9](#_bookmark24) |
| [5.1 Performance Requirements](#_bookmark25) | | [10](#_bookmark25) |
| [5.2 Safety Requirements](#_bookmark26) | | [10](#_bookmark26) |
| [SR-1: Consistency: Checking the fact that all clients must be attached to one server, so](#_bookmark27) | | [10](#_bookmark27) |
| [there is an appropriate control of the information.](#_bookmark28) | | [10](#_bookmark28) |
| [5.3 Software Quality Attributes](#_bookmark29) | | [10](#_bookmark29) |
| [5.4 Business Rules](#_bookmark30) | | [10](#_bookmark30) |
| [6.](#_bookmark31) | [Other Requirements](#_bookmark31) | 11 |

1. USE CASE DIAGRAM
2. Activity Diagram 12
3. Class Diagram 15
4. Component Diagram 20
5. SEQUENCE DIAGRAME 22
6. Test Case 26
7. Practical-1 27
8. Practical-2 29
9. Practical -3 31
10. Practical-4 33
11. Practical-5 35
12. Practical-6 36
13. Practical-7 38
14. Practical-8 40
15. Practical-9 41
16. Practical-10 42
17. Practical-11 43
18. Screenshot 44

### Introduction

**The project ONGRO has been developed on MERN. The main aim for developing this project where customer can purchase and order groceries online. The system is very useful for customer can easily but the grocery products from home through internet.**

### The system reduces lot of work load for customer as well as owner. By this system online grocery shopping the product is directly delivered to customer address.

#### Purpose

This document is meant to set forth the features and requirements of Online grocry Delivery Platform

#### Document Conventions

* + 1. SRS: System Requirement Specifications

1. GUI: Graphical User Interface
2. UC: Use Case
3. Approx.: Approximately

#### Product Scope

As this is generic software it can be used by a wide variety of outlets (Retailers) to automate the process of manually maintaining the records related to the subject of maintaining the stock and cash flows.

This project is basically updating the manual chemist inventory System to Automated inventory system, so that organization can manage their record in efficient and organized form . It provides the facility to the user that the user can order medicine at any time and from any medical store near his home and he will get the medicine within an hour. This application helps those customers who cannot walk for long Or is not in a condition to walk and Those who does not have time to bring medicine from medical shop.

### Overall Description

#### Product Functions

#### 

#### Order module

#### 

#### Admin can manage the order

#### Admin can edit/delete the order

#### Admin can see the list of all order

#### Customer can see the order

#### 2. Customer module

#### Customer will be able to see his details

#### Customer will be able to update his details

#### Admin can add new customer

#### Only admin can edit , update and delete the record of the customer

**3. Product module**

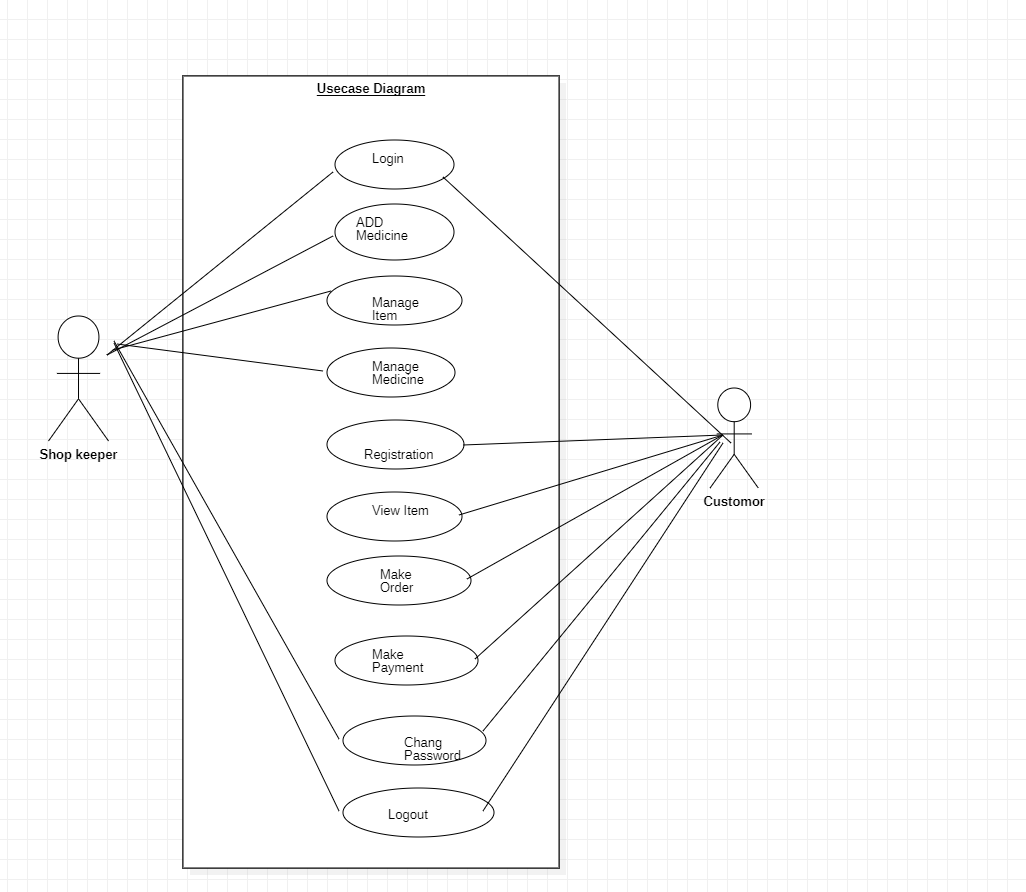
* Admin can manage the product
* Admin can edit/delete the product
* Admin can see the list of all product
* Customer can see the product

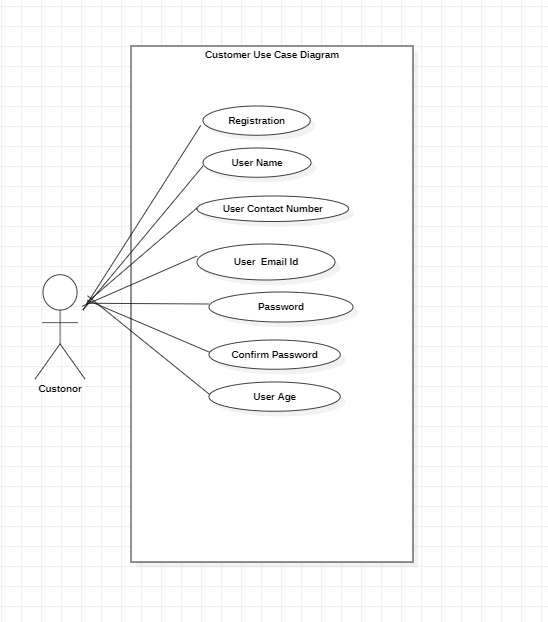
**4. Product stock module**

* Admin can manage the product stock
* Admin can edit/delete the product stock
* Admin can see the list of all product stock
* Customer can see the product stock

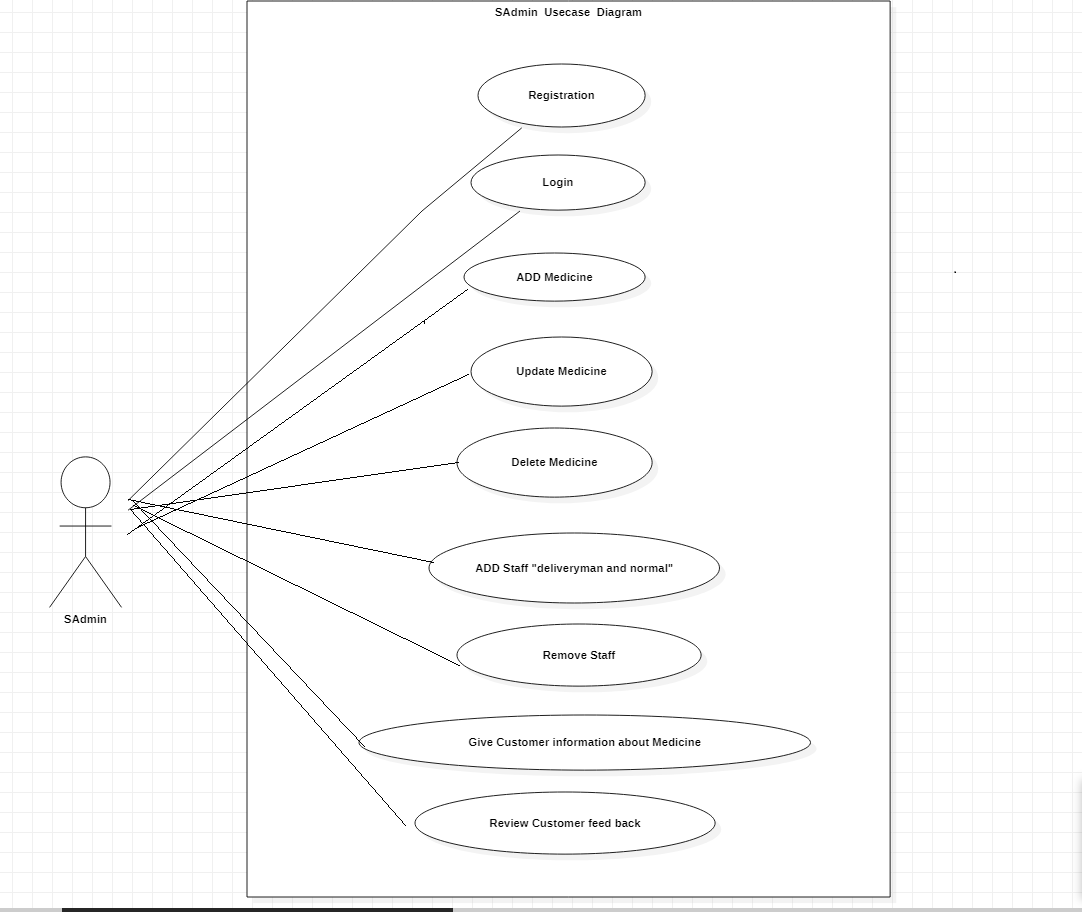
#### USE CASE DIAGRAME: A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.





CUSTOM



Shop

Activity Diagram

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

**Activities**

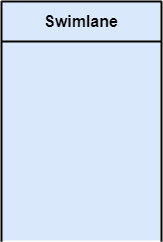
The categorization of behavior into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes.

The control flow of activity is represented by control nodes and object nodes that illustrates the objects used within an activity. The activities are initiated at the initial node and are terminated at the final node.



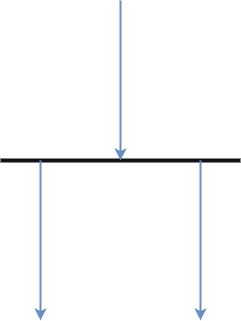
**Activity partition /swimlane**

The swimlane is used to cluster all the related activities in one column or one row. It can be either vertical or horizontal. It used to add modularity to the activity diagram. It is not necessary to incorporate swimlane in the activity diagram. But it is used to add more transparency to the activity diagram.



**Forks**

Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters. Whenever a data is received at an inward edge, it gets copied and split crossways various outward edges. It split a single inward flow into multiple parallel flows.







i nval id

valid









Eh E

# UML Class Diagram

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

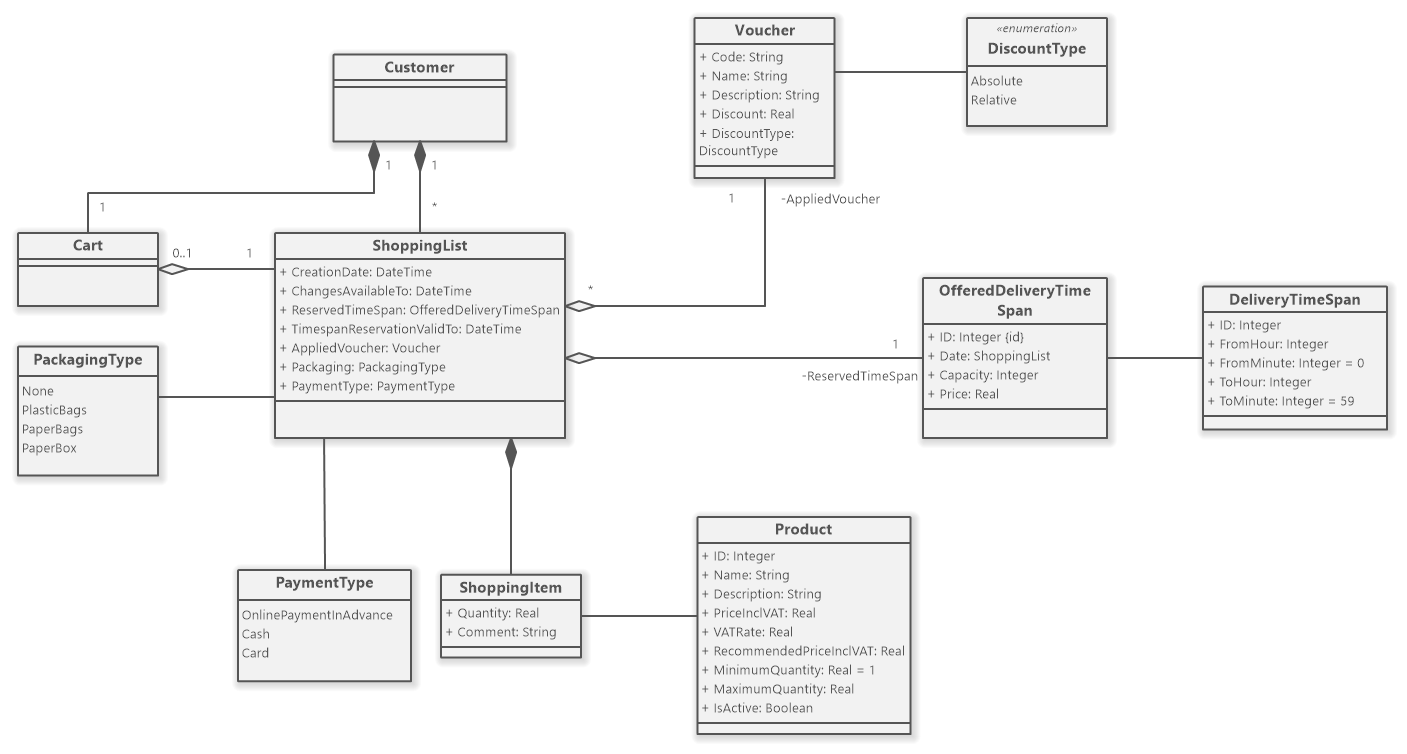
Purpose of Class Diagrams

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:

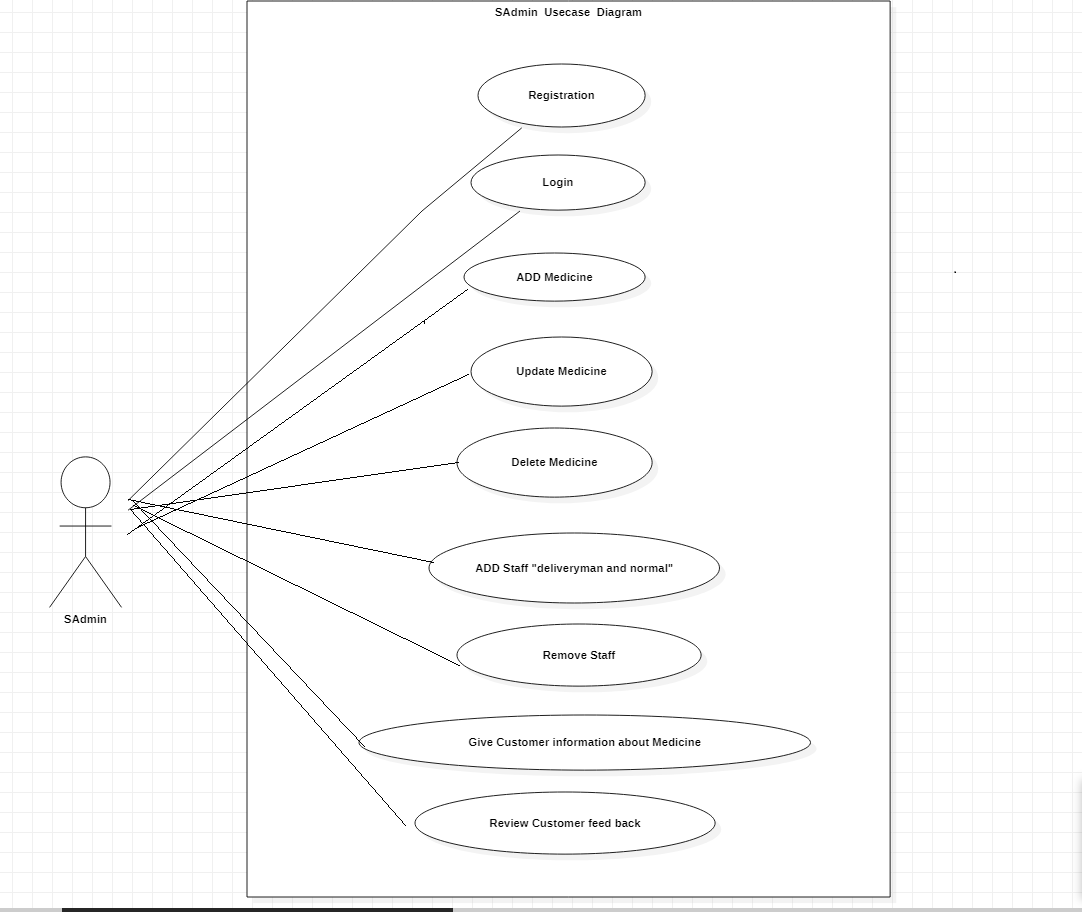
1. It analyses and designs a static view of an application.
2. It describes the major responsibilities of a system.
3. It is a base for component and deployment diagrams.
4. It incorporates forward and reverse engineering.

Benefits of Class Diagrams

1. It can represent the object model for complex systems.
2. It reduces the maintenance time by providing an overview of how an application is structured before coding.
3. It provides a general schematic of an application for better understanding.
4. It represents a detailed chart by highlighting the desired code, which is to be programmed.
5. It is helpful for the stakeholders and the developers.



CLASS DIAGRAME



Shop

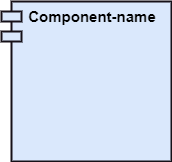
# UML Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

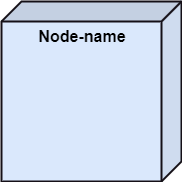
It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

Notation of a Component Diagram

1. A component



1. A node



# Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of a Sequence Diagram

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

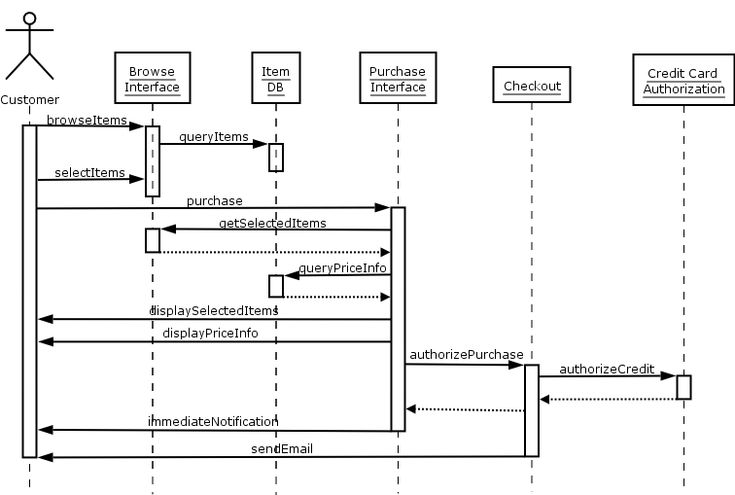
Benefits of a Sequence Diagram

1. It explores the real-time application.
2. It depicts the message flow between the different objects.
3. It has easy maintenance.
4. It is easy to generate.
5. Implement both forward and reverse engineering.
6. It can easily update as per the new change in the system.

## The drawback of a Sequence Diagram

1. In the case of too many lifelines, the sequence diagram can get more complex.
2. The incorrect result may be produced, if the order of the flow of messages changes.
3. Since each sequence needs distinct notations for its representation, it may make the diagram more complex.
4. The type of sequence is decided by the type of message.

**Sequence Diagram**



SEQUENCE DIAGRAM 2

###### Test Case:

**Practical-1**

**Aim:** Prepare an SRS document in line with the IEEE recommended standards for the specified Case Study. (Functional Requirements).

A functional requirement document helps you to define the functionality of a system or one of its subsystems. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.

**Hardware Interfaces Software Interfaces**

MEDES is a multi-user environment, it uses MERN for the web pages and MongoDB as the backend application tool.

**Communications Interfaces**

User can communicate with the admin via email provided on the website and hence uses simple mail transfer protocol (SMTP).

**System Features**

#### Login/Signup Page

Description and Priority

Provides the user with a page to login or register for the MEDES Priority = 10

Stimulus/Response Sequences Stimulus: User clicks on Login Link. Response: Login Page is displayed

Stimulus: User Enters Username and Password

Response: Username and Password are validated from MongoDB Database.

Stimulus: User Clicks on Login Button

Response: Home Page is displayed if Username and Password is correct

else Error Message is displayed and the user is asked to enter username and password again.

Stimulus: User clicks on register Link. Response: Register(signup) page is displayed

Stimulus: User enters his/her details and clicks on register button Response: User’s data gets saved; account is created and home page is

displayed

Functional Requirements

REQ-1: The user should be able to view and click on login link

REQ-2: The user should be able to enter and validate the username an

Password.

REQ-3: There should be only one account per email else and error pop-up should appear.

#### View Inventory

.

#### View User Profile

**Logout Page**

Description and Priority

**Practical-2**

**Aim:** Prepare an SRS document in line with the IEEE recommended standards for the specified Case Study. (Non-Functional Requirements).

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. They ensure the usability and effectiveness of the entire system

The Non-functional requirement of Project discussed as follows:

* 1. **Performance Requirements**
     + PE-1: Responses to activities should not take very long time onto the screen (approx.3-6s, depending upon user’s internet speed).
     + PE-2: The system should display confirmation message to the user very quickly after the user submits information to the system (depending upon user’s internet speed).

**Other Nonfunctional Requirements**

**Performance Requirements**

PE-1: Responses to activities should not take very long time onto the screen (approx.

2-5s, depending upon user’s internet speed).

PE-2: The system should display confirmation message to the user very quickly after the user submits information to the system (depending upon user’s internet speed).

PE-3: MEDES should work fine even with a number of users using concurrently.

#### Safety Requirements

SR-1: Consistency: Checking the fact that all clients must be attached to one server, so there is an appropriate control of the information.

SR-2: Users’ data should be kept confidential.

#### Software Quality Attributes

SQA-1: MEDES should be available to the users all the time.

SQA-2: The code should be neat and it should contain all the necessary comments for the ease of maintenance.

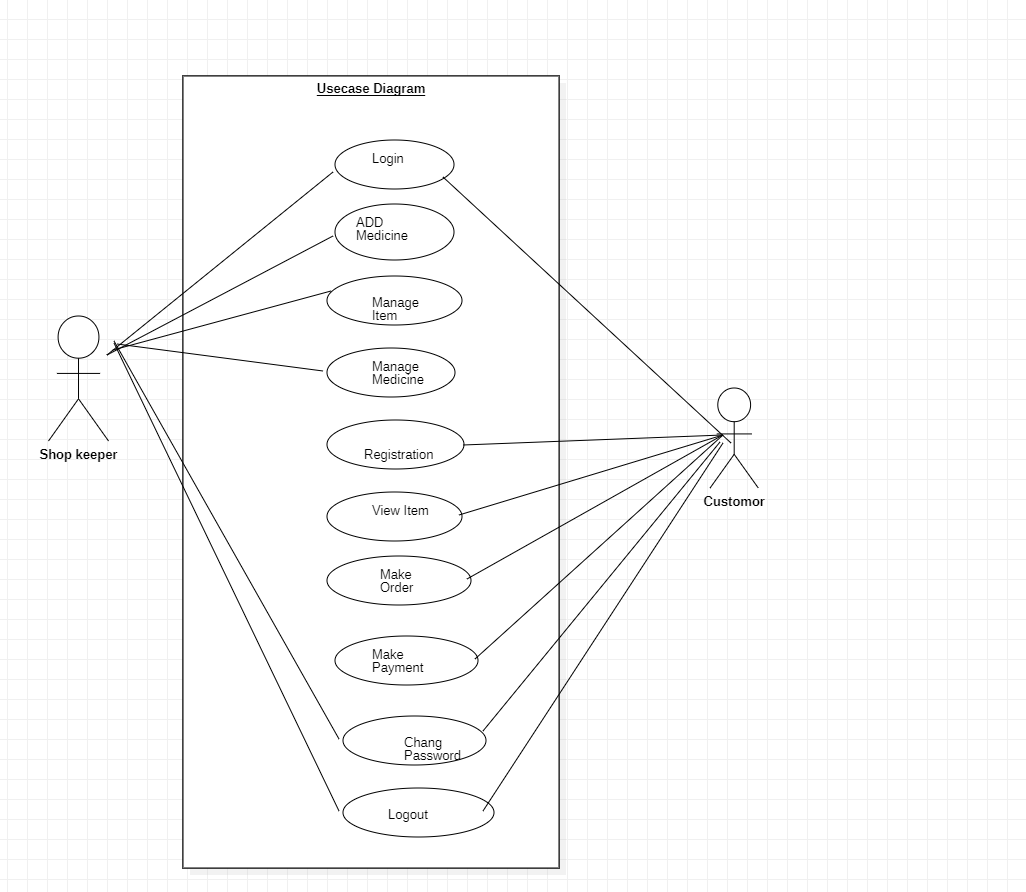
#### Business Rules

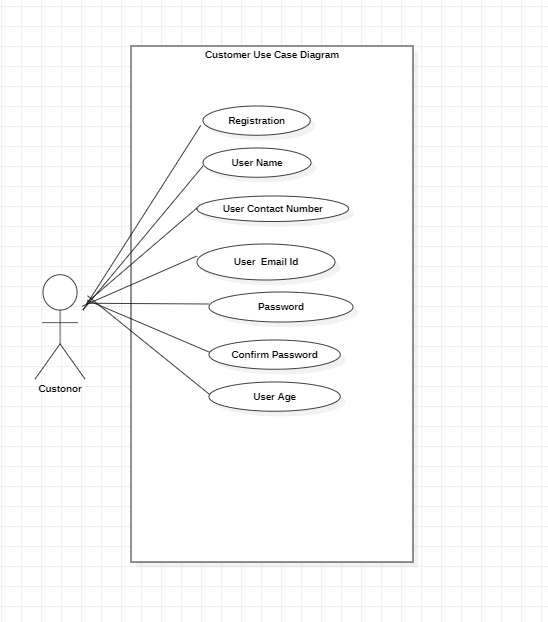
* + - * Return Policy
      * Certified Medical Store
      * Original Medicines

### Practical-3

**Aim:** Draw the use case diagram and specify the role of each of the actors for the specified Case Study of your Project.

**Use Case**: A use case is a methodology used in system analysis to identify, clarify and organize system requirements





CUSTOM

**Practical-4**

**Prepare state the precondition, post condition and function of each use case for the specified Case Study.**

**Pre Condition:** A *precondition* is a prerequisite. It’s the thing that has to happen before something else happens. For example, as a *precondition* to getting your allowance, you might have to give the dog a hairdo once a week.

**Post Condition:** Post-conditions are the things that must be true after the method is complete. The method tells clients "this is what I promise to do for you".

1. **Registration page**

###### Preconditions:

* + 1. User should have a valid email address
    2. User shouldn’t be already registered
    3. User should have a stable Internet connection
    4. Username should not have any special characters other than ‘\_’
  1. **Postconditions:**
     1. User should get a verification code on their email
     2. User’s account should be created after the verification

1. **Login Page**
   1. **Preconditions:**
      1. User should have a registered account
      2. User should enter valid username/email
      3. Users should enter the correct password
      4. Users should have a stable Internet connection
   2. **Postconditions:**
      1. User should be redirected to home page

**3Logout Page**

**Preconditions:**

* + 1. User should be logged in
    2. User should have a stable Internet connection
  1. **Postconditions:**
     1. User should be logged out from all the tabs

1. Login/register page should appearOther preconditions and postconditions

###### Preconditions:

* + 1. User should have basic knowledge of using websites
  1. **Postconditions:**
     1. User should be able to use the website easily 5 Payment Condition;
  2. Card Number must be enter
  3. CVV number must be numeric

Month and year code must be

# Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of a Sequence Diagram

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

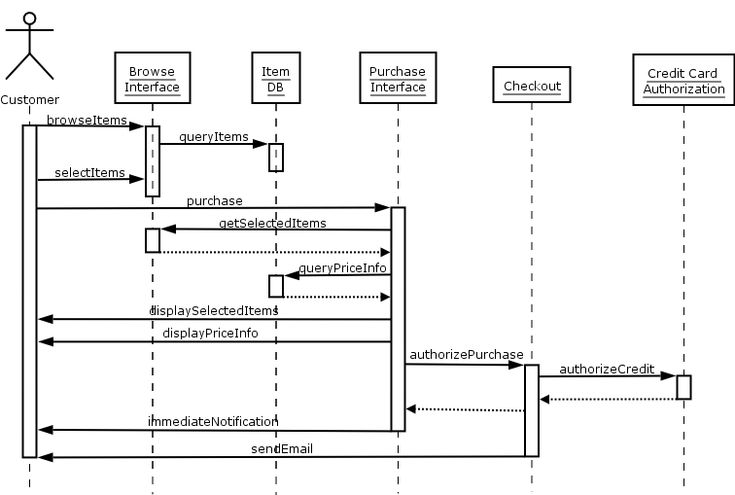
Benefits of a Sequence Diagram

1. It explores the real-time application.
2. It depicts the message flow between the different objects.
3. It has easy maintenance.
4. It is easy to generate.
5. Implement both forward and reverse engineering.
6. It can easily update as per the new change in the system.

## The drawback of a Sequence Diagram

1. In the case of too many lifelines, the sequence diagram can get more complex.
2. The incorrect result may be produced, if the order of the flow of messages changes.
3. Since each sequence needs distinct notations for its representation, it may make the diagram more complex.
4. The type of sequence is decided by the type of message.

**Sequence Diagram**



Purpose of a Sequence Diagram

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

Benefits of a Sequence Diagram

1. It explores the real-time application.
2. It depicts the message flow between the different objects.
3. It has easy maintenance.
4. It is easy to generate.
5. Implement both forward and reverse engineering.
6. It can easily update as per the new change in the system.

## The drawback of a Sequence Diagram

1. In the case of too many lifelines, the sequence diagram can get more complex.
2. The incorrect result may be produced, if the order of the flow of messages changes.
3. Since each sequence needs distinct notations for its representation, it may make the diagram more complex.

The type of se

SEQUENCE DIAGRAM 2

###### Test Case:

**Practical-1**

**Aim:** Prepare an SRS document in line with the IEEE recommended standards for the specified Case Study. (Functional Requirements).

A functional requirement document helps you to define the functionality of a system or one of its subsystems. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.

**Hardware Interfaces Software Interfaces**

MEDES is a multi-user environment, it uses MERN for the web pages and MongoDB as the backend application tool.

**Communications Interfaces**

User can communicate with the admin via email provided on the website and hence uses simple mail transfer protocol (SMTP).

**System Features**

#### Login/Signup Page

Description and Priority

Provides the user with a page to login or register for the MEDES Priority = 10

Stimulus/Response Sequences Stimulus: User clicks on Login Link. Response: Login Page is displayed

Stimulus: User Enters Username and Password

Response: Username and Password are validated from MongoDB Database.

Stimulus: User Clicks on Login Button

Response: Home Page is displayed if Username and Password is correct

else Error Message is displayed and the user is asked to enter username and password again.

Stimulus: User clicks on register Link. Response: Register(signup) page is displayed

Stimulus: User enters his/her details and clicks on register button Response: User’s data gets saved; account is created and home page is

displayed

Functional Requirements

REQ-1: The user should be able to view and click on login link

REQ-2: The user should be able to enter and validate the username an

Password.

REQ-3: There should be only one account per email else and error pop-up should appear.

#### View Inventory

.

#### View User Profile

**Logout Page**

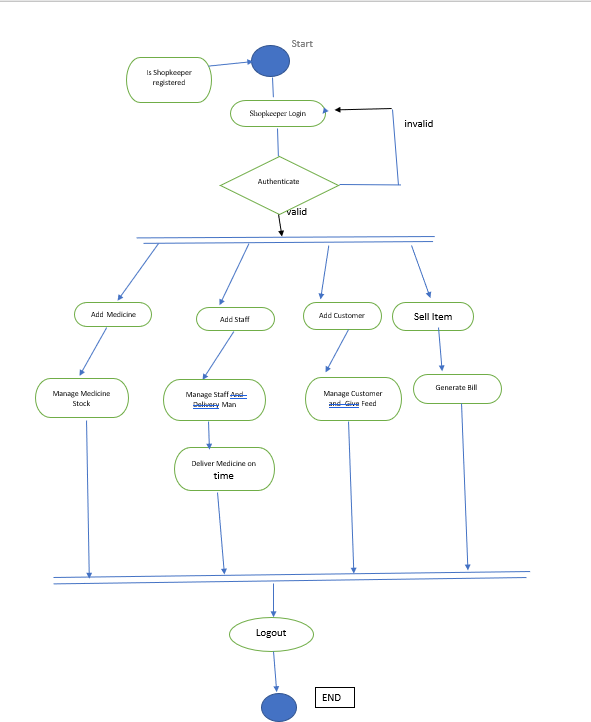
Description and Priority

* 1. number

**Practical-5**

**Draw the activity diagram for the specified Case Study.**

**Activity Diagram:** An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



### Practical-6

###### Aim: Identify the classes. Classify them as weak and strong classes and draw the class diagram for the specified Case Study.

**Class Diagram: Class diagram is a static diagram. ... The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and**

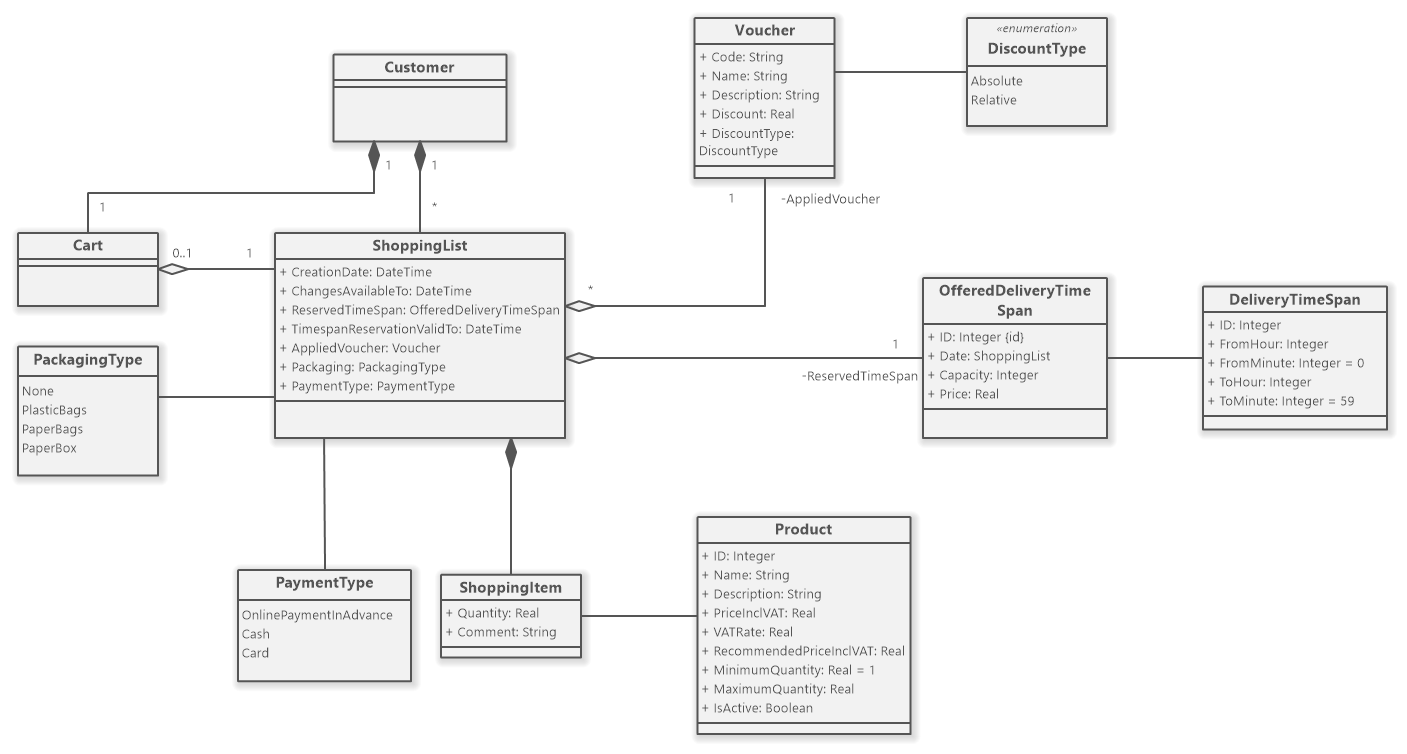
Purpose of Class Diagrams

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:

* + 1. It analyses and designs a static view of an application.
    2. It describes the major responsibilities of a system.
    3. It is a base for component and deployment diagrams.
    4. It incorporates forward and reverse engineering.

## Benefits of Class Diagrams

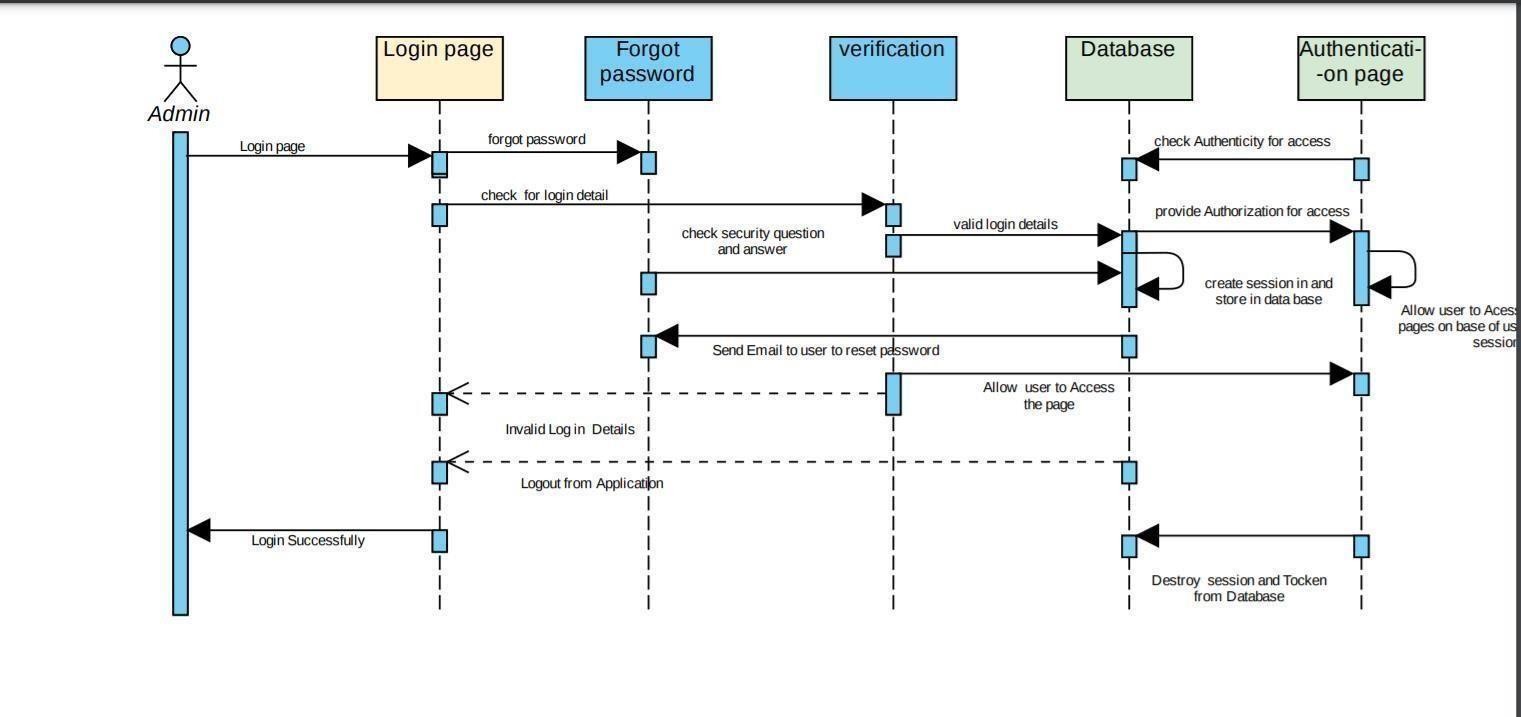
1. It can represent the object model for complex systems.
2. It reduces the maintenance time by providing an overview of how an application is structured before coding.
3. It provides a general schematic of an application for better understanding.
4. It represents a detailed chart by highlighting the desired code, which is to be programmed.
5. It is helpful for the stakeholders and the developers.

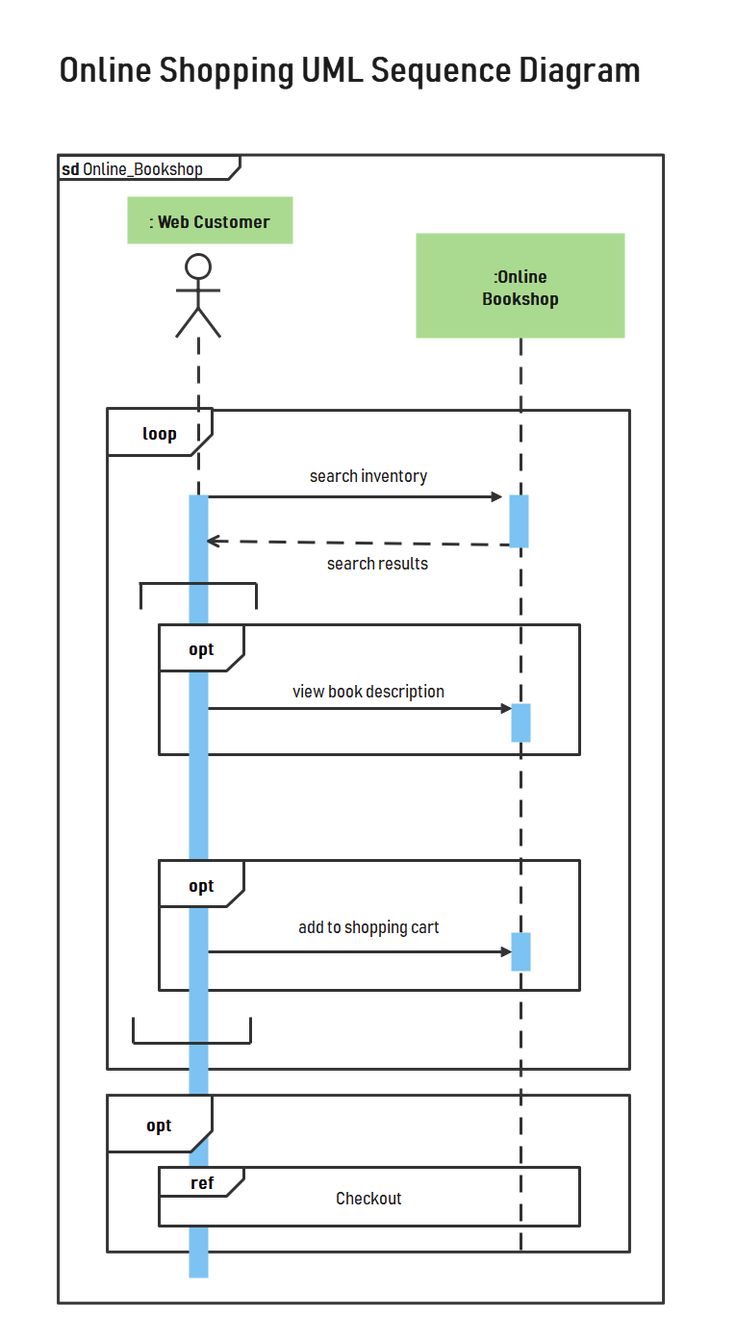


### Practical-7

###### Aim: Draw the sequence diagram for any two scenarios.

**Sequence Diagram:** A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario.

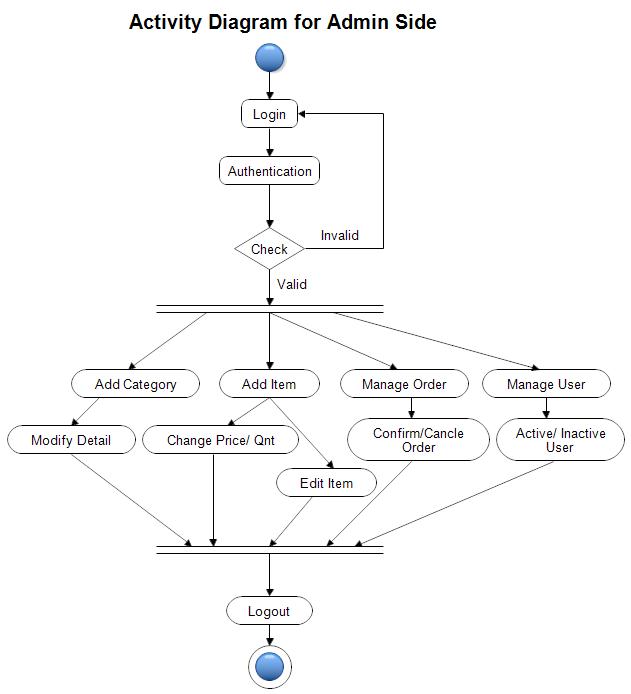




**Practical-8**

**Aim: Draw the collaboration diagram for the specified Case Study.**

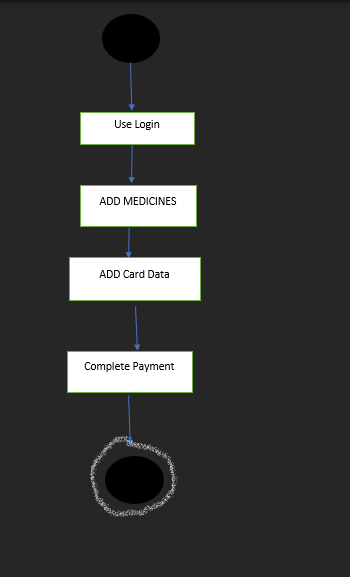
**Component Diagram:** A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.



**Practical-9**

**Aim: Draw the state chart diagram for the specified Case Study.**

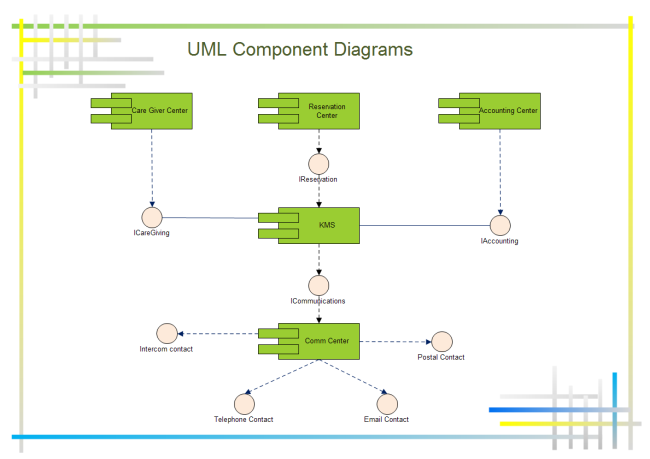
**State Chart Diagram:** Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination



**Practical-10**

**Aim: Draw the component diagram for the specified Case Study.**

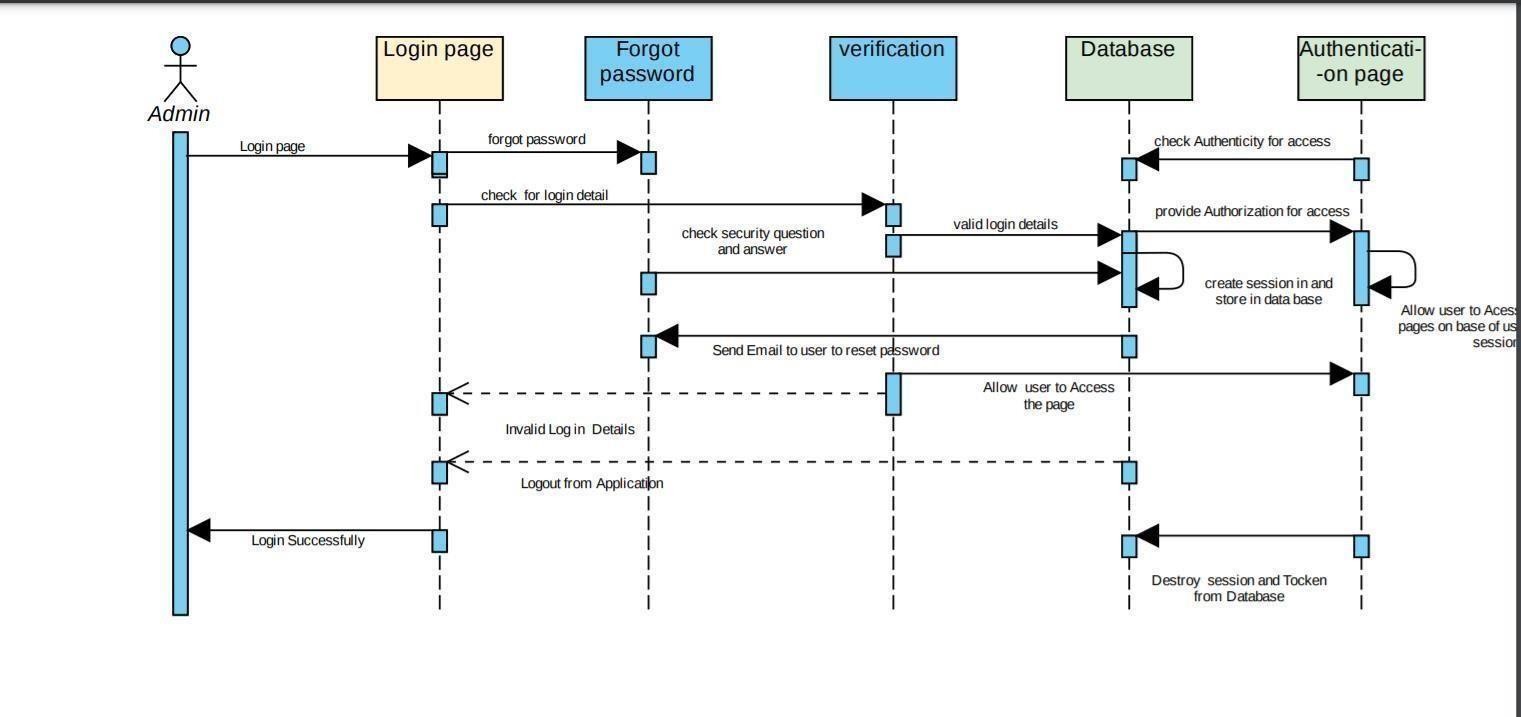
**Component Diagram:** A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node

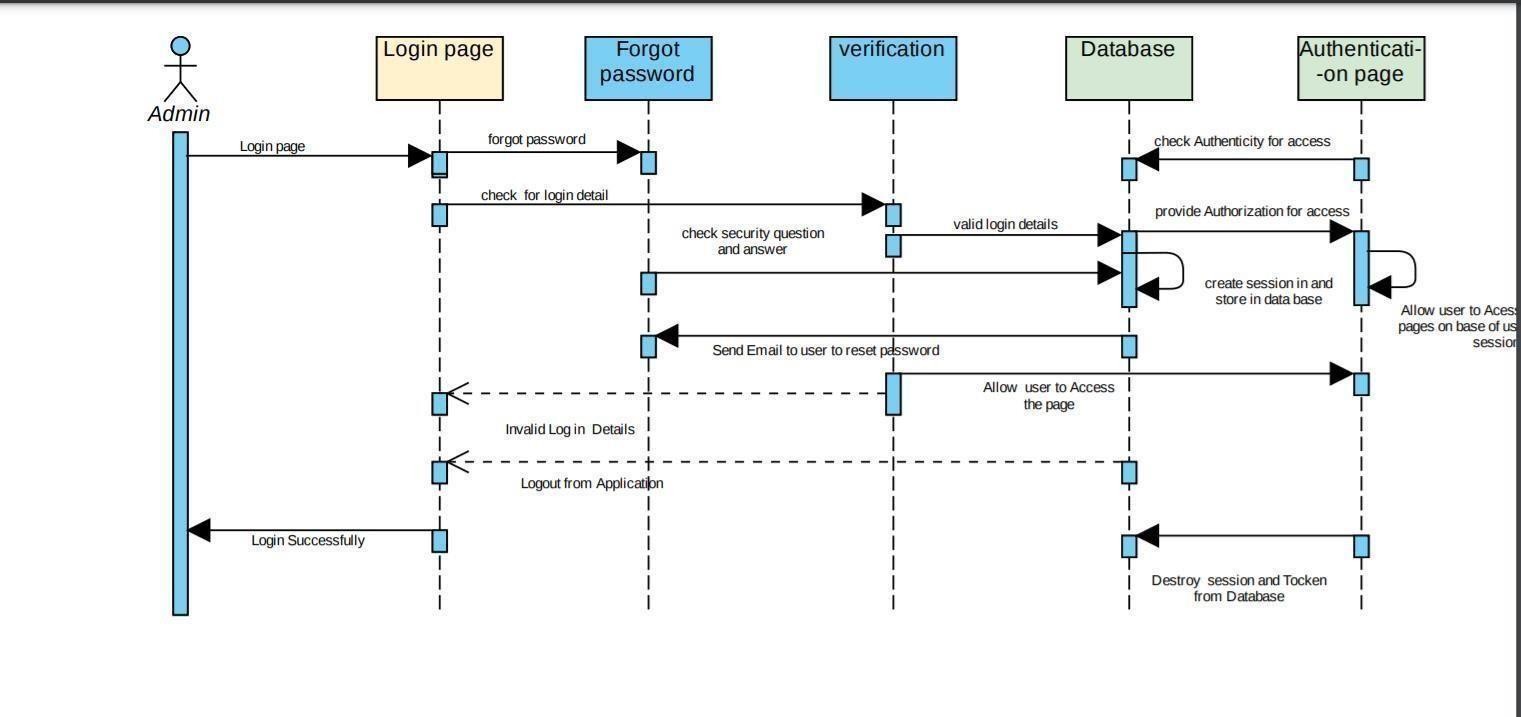


**Practical-11**

**Aim: Design a test suite for the specified Case Study.**

**Test Cases:** A test case is exactly what it sounds like: a test scenario measuring functionality across a set of actions or conditions to verify the expected result.



PROJECT SCREENSHOT: