CANTEEN MANAGEMENT SYSTEM

SUBJECT :OODP

SUBJECT CODE :18CSC202J & 18AIC203J

TEAM MEMBERS :

K.DEEKSHITHA REDDY - RA2111026010519
K.RITISH REDDY - RA2111033010153

INDEX

PROBLEM DESCRIPTION

Main objective is to build the system to provide ordering and reservation service online to the customer. The UML diagrams and code generations we provide will help to better understand the process and roles behind the system.

AIM:

- To provide Admin Users can search Food Items, view description of a selected Food Item, add Food Item, update Food Item and delete Food Item.
- It shows the activity flow of editing, adding and updating of Canteen**.**

1)USE CASE DIAGRAM :

USE CASE DIAGRAM on CANTEEN MANAGEMENT

SYSTEM: **EXPLANATION:**

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

USED TOOLS :

- ACTOR
- USE CASE
- ASSOCIATION
- INCLUDE

We used 4 actors here namely customer, manager, chef, and waiter and we used several use cases like search menu, select item, add item, etc.

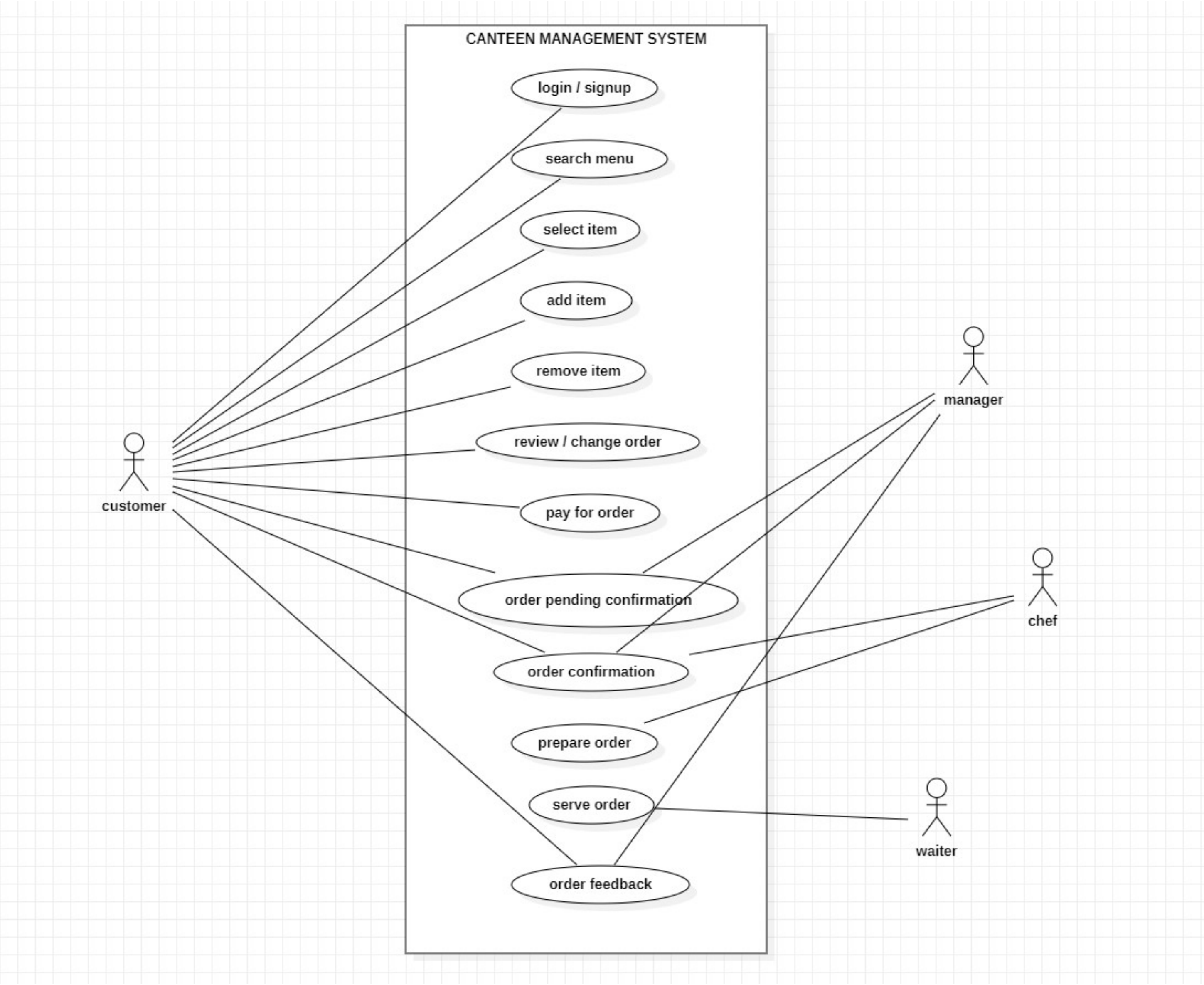
Report:

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

An effective use case diagram can help our team discuss and represent scenarios in which our system interacts with people, organizations, or external systems. Goals that your system helps those entities (actors) achieve.

Use-case diagrams illustrate and define the context of the entire part of the system. We can model a complex system with a single use-case or many use-case diagrams
In our CANTEEN MANAGEMENT SYSTEM, it is very clear about the work done by the actors

Hence the Use case Diagram on the Canteen Management system was done successfully.


CANTEEN MANAGEMENT SYSTEM
login / signup
search menu
select item
add item
remove item
review / change order
pay for order
order pending confirmation
order confirmation
prepare order
serve order
order feedback
customer
manager
chef
waiter

2) CLASS DIAGRAM:

CLASS DIAGRAM ON CANTEEN MANAGEMENT SYSTEM :

Explanation:

Class diagrams are the blueprints of your system. We use class diagrams to model the objects which make up the system, and to display the relationships between the objects. They describe what their objects do and the services that they provide. Class diagrams are useful in many stages of system design. In the class diagram, the class depicts information and behavior. Consists of 3 sections

top: hold class name
middle: attributes

lower: holds operations or methods

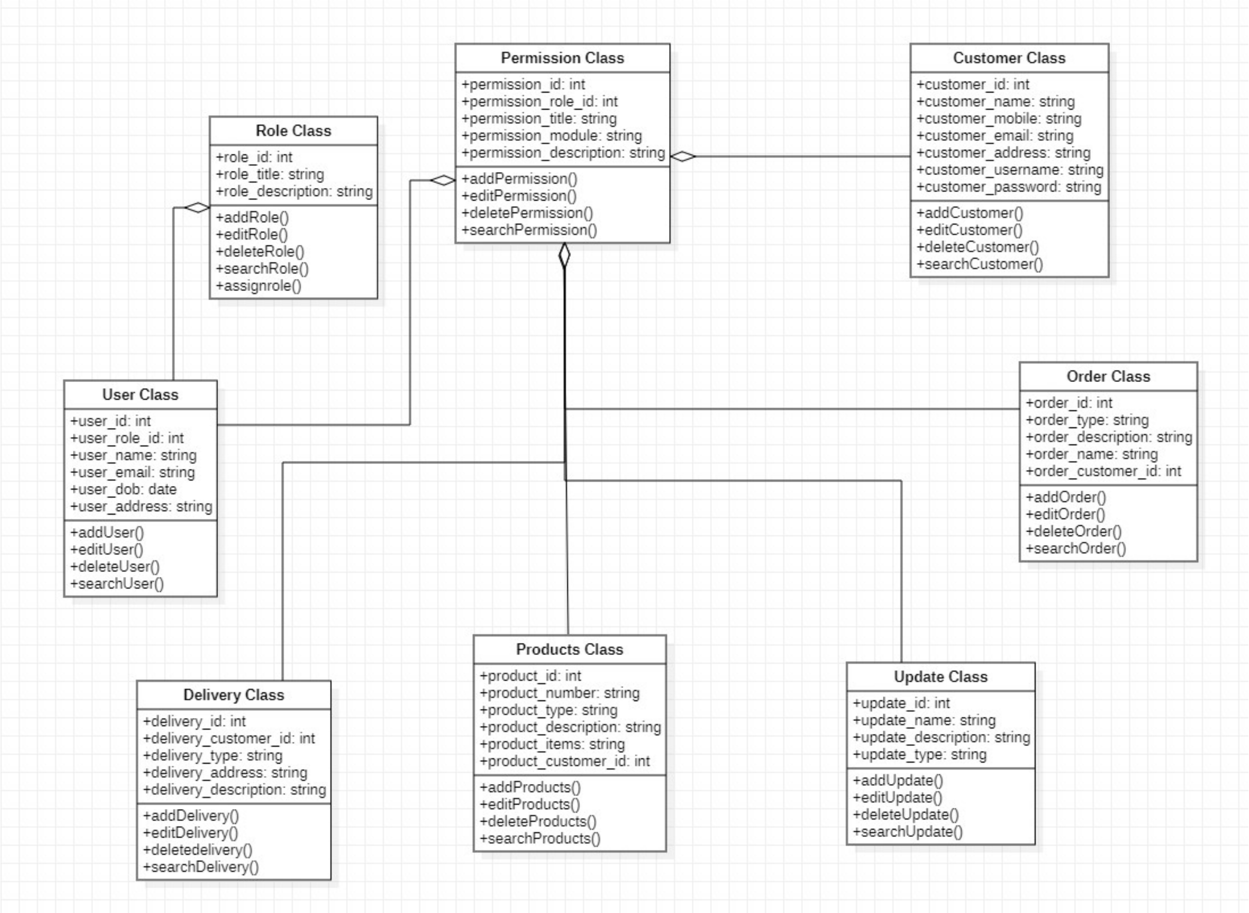
Used Tools:

- class
- association
- generalization

To begin with, we used a class diagram and named the top section Customer and gave the attributes in the middle section Methods. In the bottom section we repeated this for 7 more classes and gave various names, attributes, and methods to it.

We used the Association tool to connect classes

At last, we generated C++ code of what we have done.



REPORT WRITING:

Class diagrams Illustrate data models for information systems, no matter how simple or complex. Better understand the general overview of the schematics of an application. Visually express any specific needs of a system and disseminate that information throughout the business.

Our diagram Illustrates the structure of a model by using attributes, operations, and signals.

Hence the class diagram for the CANTEEN MANAGEMENT SYSTEM was performed successfully.

3)SEQUENCE DIAGRAM

SEQUENCE DIAGRAM on Canteen Management System:

Explanation:

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. The main purpose of a sequence diagram is to define event sequences that result in some desired outcome.

The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimension shows, top-down, the time sequence of messages/calls as they occur, and the horizontal dimension shows, left to right, the object instances that the messages are sent to.

Lifelines

When drawing a sequence diagram, lifeline notation elements are placed across the top of the diagram. Lifelines represent either roles or object instances that participate in the sequence being modeled.

Here we have used five lifelines namely user, login success, canteen management, order management, and bill management.

Messages

The first message of a sequence diagram always starts at the top and is typically located on the left side of the diagram for readability. There are many kinds of messages like synchronous, asynchronous, self, etc.

In this diagram, nearly 13 messages are used. 2 types of messages are used namely,

MESSAGE
SELF MASSAGE

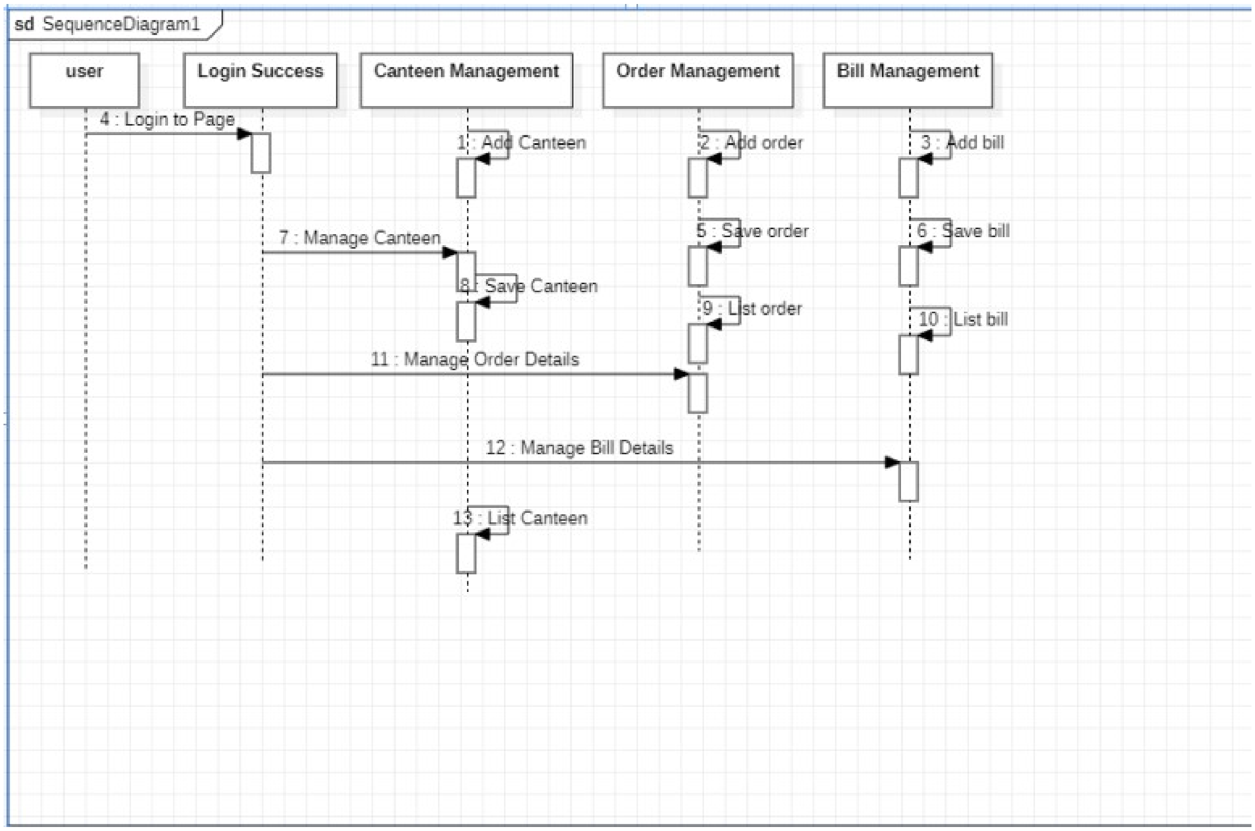
To draw an interaction diagram, we must identify

1) Objects in the interaction

2) Message flowing among the objects

3) Sequence in which messages flow

4) Object organization


sd SequenceDiagram1
user
Login Success
Canteen Management
Order Management
Bill Management
4 : Login to Page
1 : Add Canteen
2 : Add order
3 : Add bill
5 : Save order
6 : Save bill
7 : Manage Canteen
8 : Save Canteen
9 : List order
10 : List bill
11 : Manage Order Details
12 : Manage Bill Details
13 : List Canteen

4)ACTIVITY DIAGRAM

ACTIVITY DIAGRAM ON CANTEEN MANAGEMENT SYSTEM: Explanation: An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram.

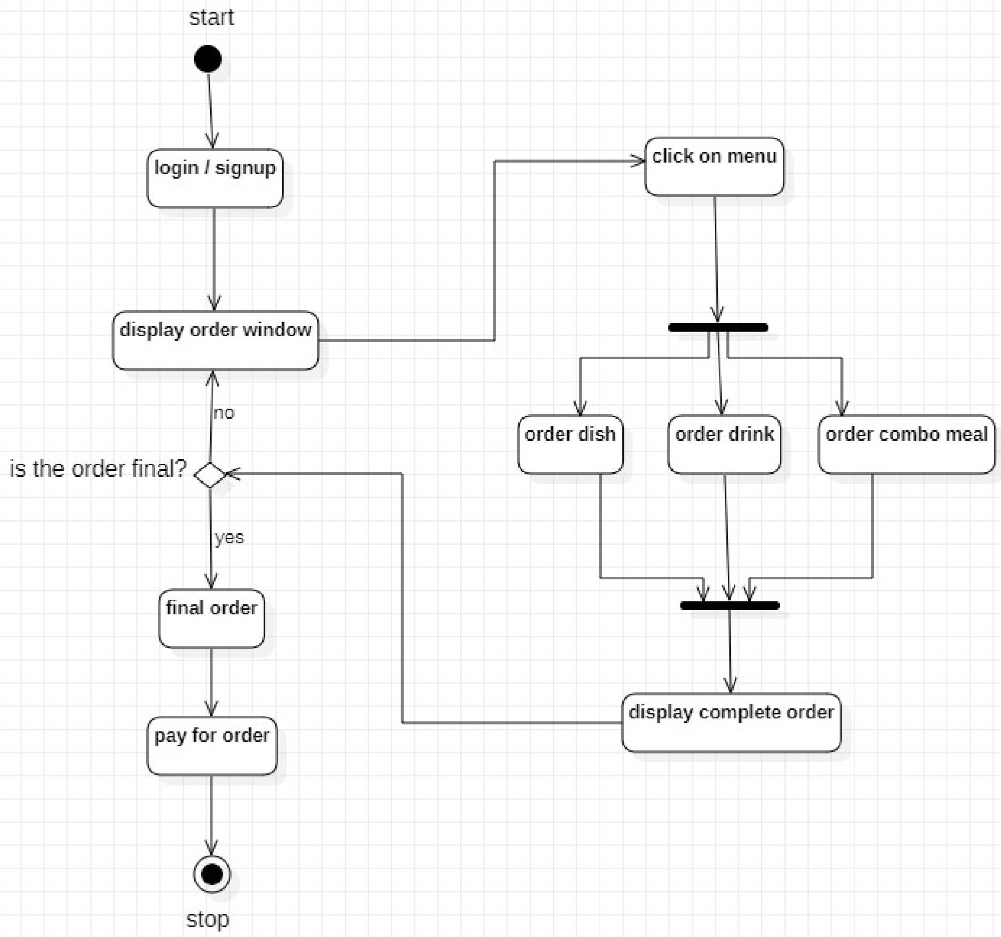
TOOLS:

- INITIAL
- ACTION
- FINAL
- CONTROL FLOW
- MERGE
- DECISION
- FORK NODE
- JOIN NODE

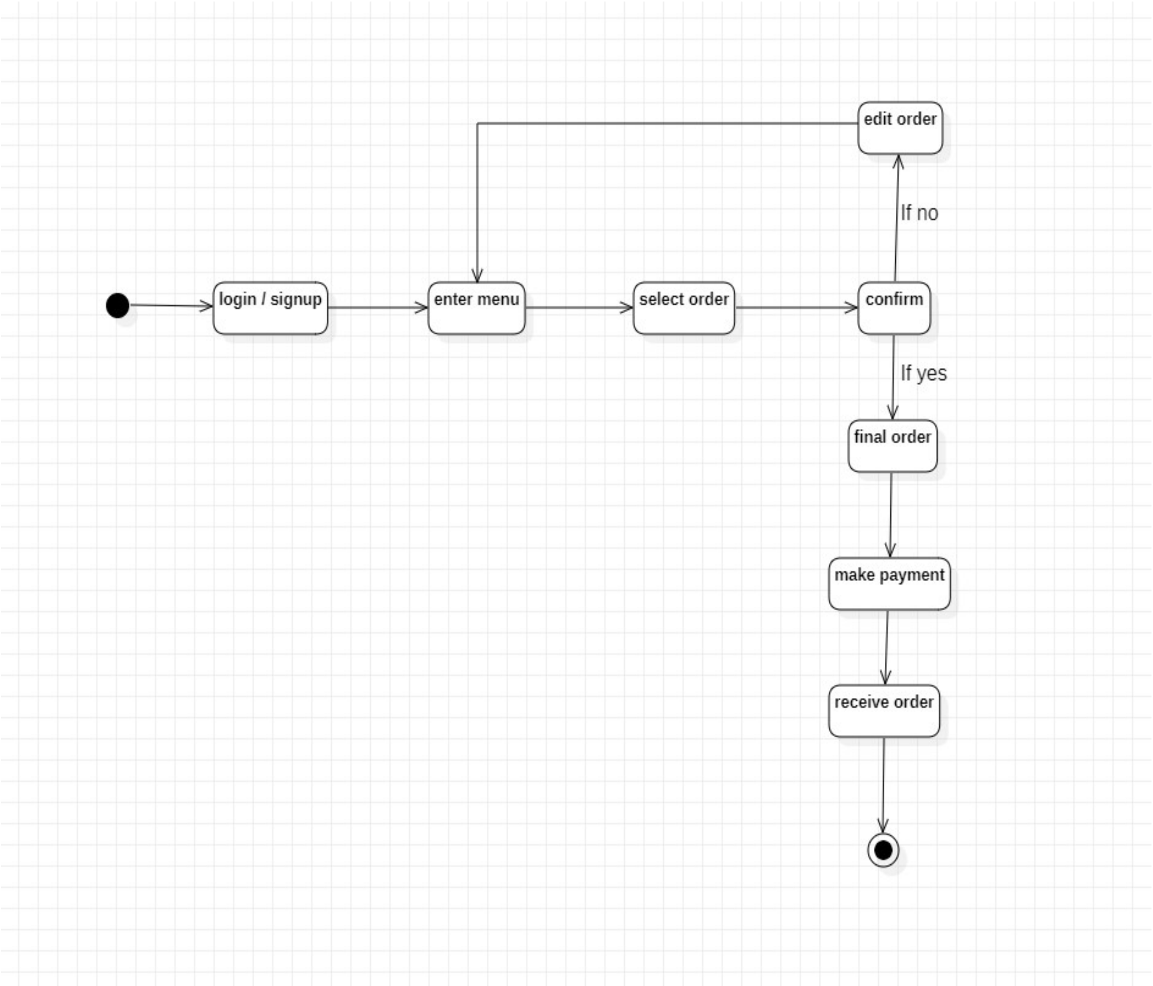
Starting from the beginning we used actions which are connected with control flow and used a decision tool for taking decisions and a fork node to divide one action into two and join nodes to combine 2 actions from the fork node and finally we ended the diagram with Final tool.

REPORT:

Activity diagrams describe the steps performed in a UML use case. Illustrate a business process or workflow between users and the system. Simplify and improve any process by clarifying complicated use cases. Model software architecture elements, such as method, function, and operation. We successfully performed and executed the ACTIVITY DIAGRAM for CANTEEN MANAGEMENT SYSTEM.


start
login / signup
display order window
click on menu
order dish
order drink
order combo meal
display complete order
is the order final?
no
yes
final order
pay for order
stop

5) STATE CHART DIAGRAM :STATE CHART DIAGRAM ON CANTEEN MANAGEMENT SYSTEM



EXPLANATION:

State diagrams enable you to describe the behavior of objects during their entire life span. In addition the different states and state changes as well as events causing transactions can be described .

It depicts the whole process of food order process in the

canteen via online. NOTATIONS USED :

- initial state
- state
- final state
- transition

Starting with the initial state we created 8 states which perform various various actions and they are connected with control flow.

REPORT:

The above state chart diagram depicts the whole food order process via online canteen in a visual representation.
In general,illustrating use case scenarios in a business context describing how an object moves through various states within a lifetime

Hence we successfully performed STATE CHART DIAGRAM ON CANTEEN management system using STARUML

6.PACKAGE DIAGRAM :

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages. A package is a grouping of relatedUML elements, such as diagrams, documents, classes, or even other packages. Each element is nested within the package, which is depicted as a file folder within the diagram, then arranged hierarchically within the diagram. Package diagrams are most commonly used to provide a visual organization of the layered architecture within any UML classifier, such as a software system.

Benefits of a package diagram :

- A well-designed package diagram provides numerous benefits to those looking to create a visualization of their UML system or project.
- They provide a clear view of the hierarchical structure of the various UML elements within a given system.
- These diagrams can simplify complex class diagrams into well-ordered visuals.
- They offer valuable high-level visibility into large-scale projects and systems.
- Package diagrams can be used to visually clarify a wide variety of projects and systems.
- These visuals can be easily updated assystems and projects evolve.

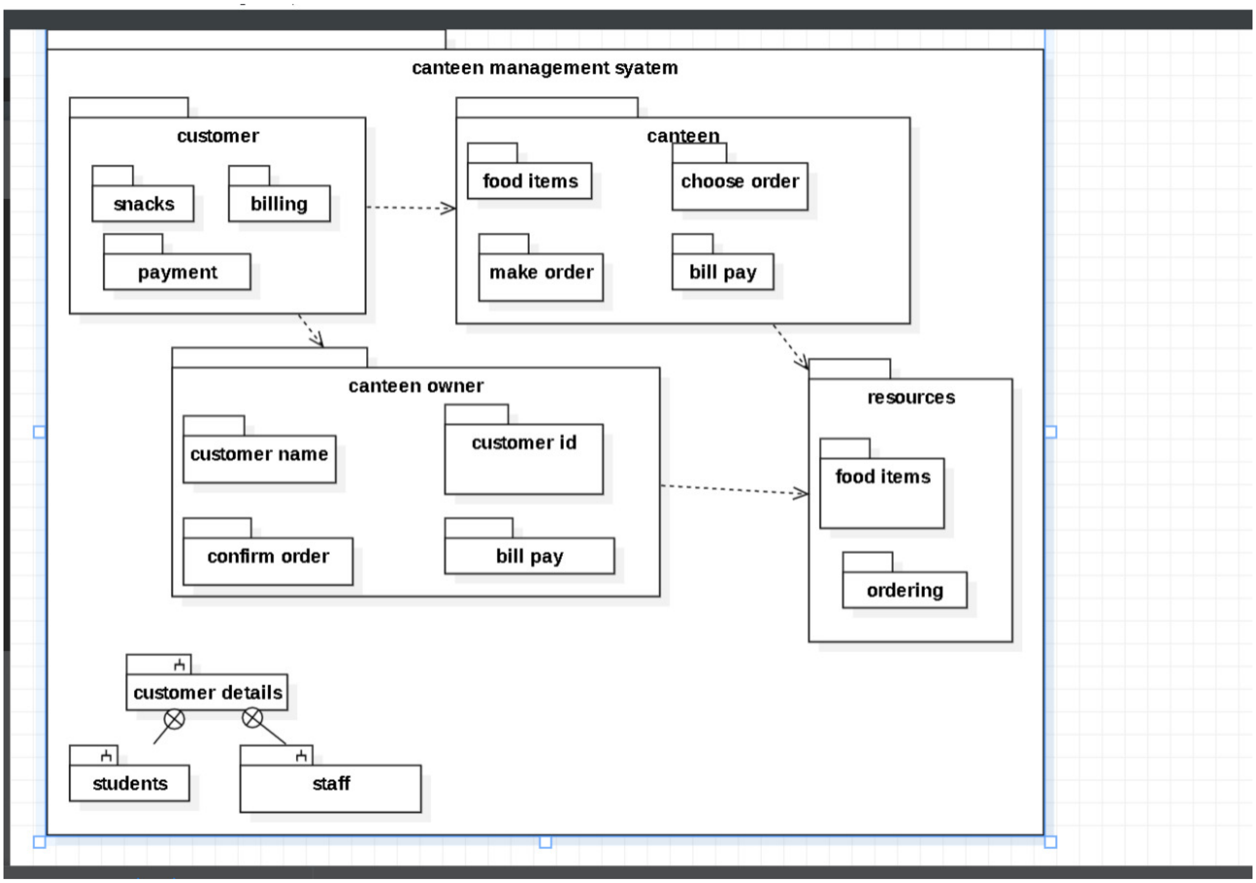
Basic components of a package diagram

The makeup of a package diagram is relatively simple. Each diagram includes only two symbols:

These symbols can be used in a variety of ways to represent different iterations of packages, dependencies, and other elements within a system. Here are the basic components you'll find within a package diagram:

- **Package**: A namespace used to group together logically related elements within a system. Each element contained within the package should be a packageable element and have a unique name.

- **Packageable element**: A named element, possibly owned directly by a package. These can include events, components, use cases, and packages themselves. Packageable elements can also be rendered as a rectangle within a package, labeled with the appropriate name.
- **Dependencies**: A visual representation of how one element (or set of elements) depends on or influences another. Dependencies are divided into two groups: access and import dependencies. (See next section for more info.)
- **Element import**: A directed relationship between an importing namespace and an imported packageable element. This is used to import select individual elements without resorting to a package import and without making it public within the namespace.
- **Package import**: A directed relationship between and importing namespace and an imported package. This type of directed relationship adds the names of the members of the imported package to its own namespace
- **Package merge**: A directed relationship in which the contents of one package are extended by the contents of another. Essentially, the content of two packages are combined to produce a new package.



7.component diagram:

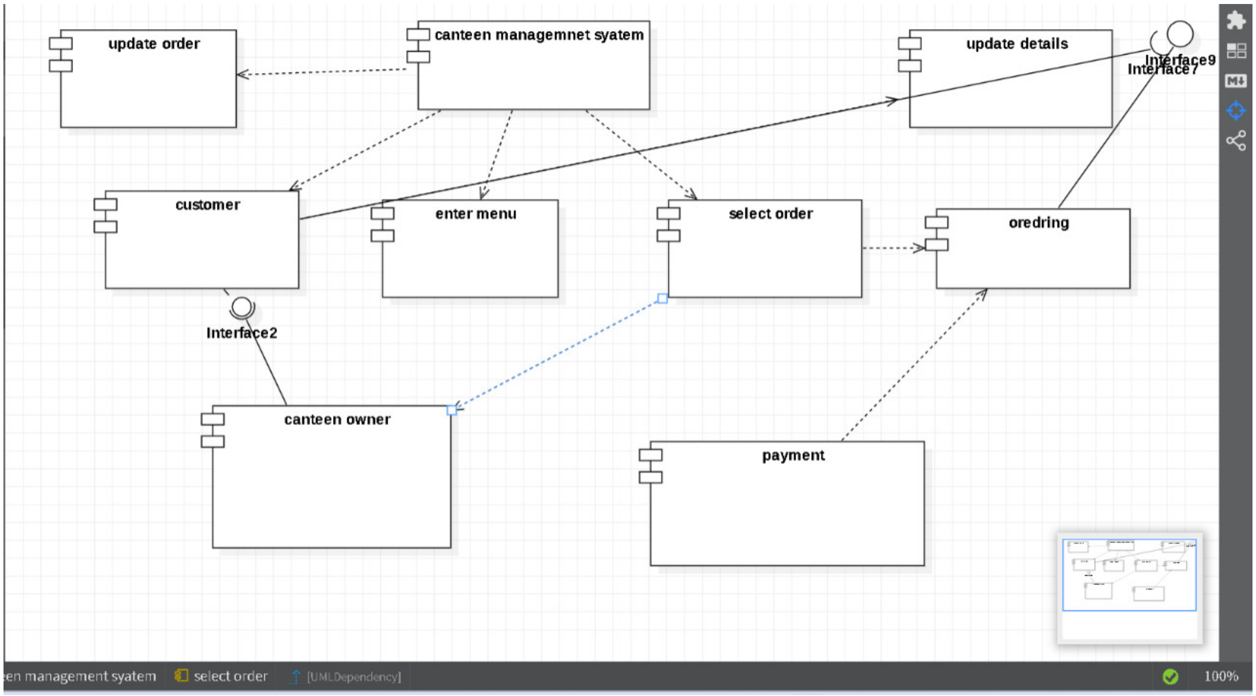
Component diagram of canteen management system which shows components , provided and required interfaces, ports, and relationships between canteen , bill employee ,\ food items,ordered products . This type of diagram is used in component based development to describe systems with service oriented architecture .Canteen management system UML component diagram describes the organization and writing of the physical components in a system.

Components of UML component diagram of canteen management system :

- **Canteen component**
- **Bill employee component**
- **Food item component**
- **Orders component**
- **Products component**

Features of canteen management system component diagram :

- You can show the models the components of the canteen management system .
- Model the database scheme of the canteen management system .
- Model the executables of an application of the canteen management system .
- Model the system source code of the canteen management system .


update order
canteen managemnet syatem
update details
Interface9
Interface7
customer
enter menu
select order
oredring
Interface2
canteen owner
payment
en management syatem
select order
[UMLDependency]
100%

8.Deployment diagram:

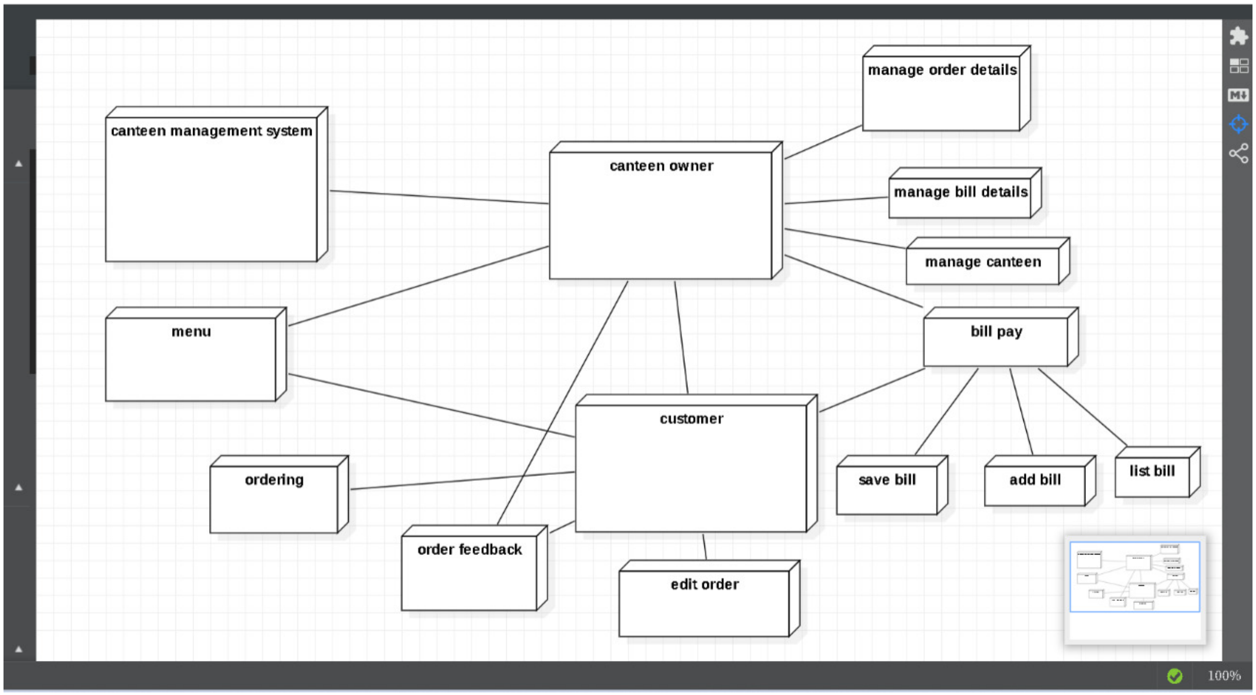
A deployment diagram in the Unified modeling language models the *physical* deployment of artifacts on nodes. To describe website, for example, a deployment diagram would show what hardware components ("nodes") exist , what software components ("artifacts'') run on each node and how the different pieces are connected

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have subnodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

There are two types of Nodes:

1. Device Node
2. Execution Environment Node

Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones. An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute another executable software elements.



Deployment diagrams are used to visualize the hardware processors/ nodes/ devices of a system, the links of communication between them and the placement of software files on that hardware.A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

9. COMMUNICATION DIAGRAM:

COLLABORATION DIAGRAM named as COMMUNICATION DIAGRAM between two objects how messages can be communicated can be clearly shown on a communication or collaboration diagram. Collaboration diagrams messages are always coming with a number In the communication diagram elements are organized with space. They are very useful for visualizing relations between objects that collaborate with each other to perform a specific task. This helps to determine the accuracy of a static model. Messages that objects send to themselves are loops. Here in our diagram, we use some tools namely LIFELINE, Forward Message, and connector.

2 Objects are used in communication diagram

1. Supplier
2. Client In this diagram numbering always start with 1 and continues accordingly

Objects : We created 4 objects using the lifeline tool with object names as user, web, worker1, and worker2.

Links: Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.

Messages: Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include numbers starting with one .

APPLICATIONS OF COLLABORATION DIAGRAMS:

i) MODELING COLLABORATIONS

ii) Visualizing the complex logic behind an operation

iii) Exhibiting many alternative scenarios for the same use case.

REPORT:

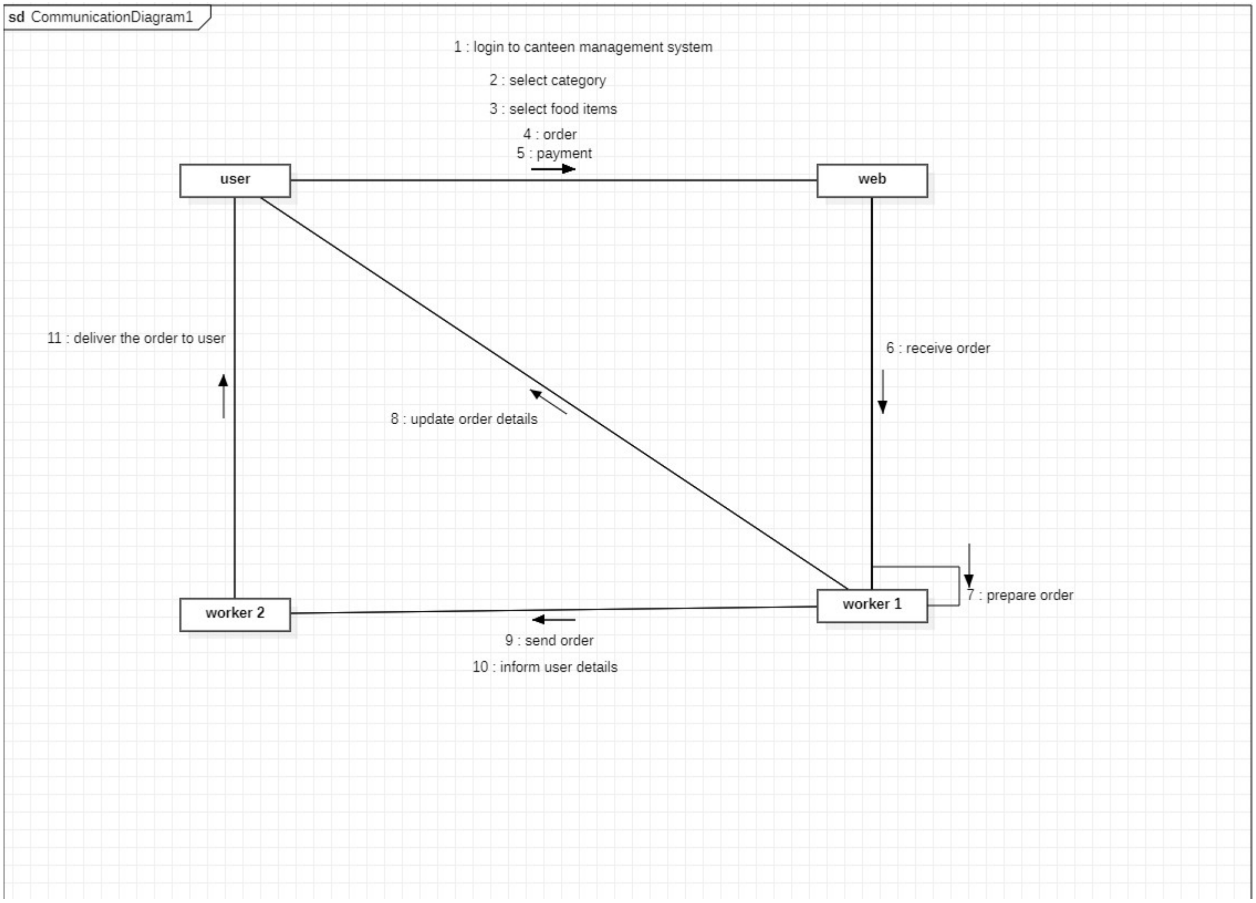
The below collaboration diagram shows the interaction clearly between user, web, worker1, worker2.

The collaboration diagram shows the behind steps of how a customer receives the order he orders online. It is much more clear because of numbers that come by default, which is a major advantage of collaboration or communication diagrams here that is starting from login to delivering the order.

UML communication diagrams, like the Sequence diagram- a kind of interaction diagram,shows how objects interact. A communication diagram is an extension of an object diagram that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, a communication diagram shows the messages the objects send each other.

Purpose of Communication Diagram

- Model message passing between objects or roles that deliver the functionalities of use cases and operations
- Model mechanisms within the architectural design of the system
- Capture interactions that show the passed messages between objects and roles within the collaboration scenario
- Model alternative scenarios within use cases or operations that involve the collaboration of different objects and interactions
- Support the identification of objects (hence classes), and their attributes (parameters of message) and operations (messages) that participate in use cases


sd CommunicationDiagram1
1 : login to canteen management system
2 : select category
3 : select food items
4 : order
5 : payment
user
web
11 : deliver the order to user
6 : receive order
8 : update order details
7 : prepare order
worker 2
worker 1
9 : send order
10 : inform user details