PRACTICAL FILE

ON

Software Engineering Lab

BCA 4th semester

Course code: BCA- 272

Session: 2022-23

SUBMITTED TO:

DR. INDU SHARMA
Assistant Professor IT

SUBMITTED BY:

Name: Harshit garg
Roll No: 00915102021
Class: BCA 4th Sem

INDEX

INTRODUCTION TO UMBRELLO

Umbrello UML Modeller is a UML diagram tool that can support you in the software development process. Especially during the analysis and design phases of this process, Umbrello UML Modeller will help you to get a high quality product. UML can also be used to document your software designs to help you and your fellow developers.

Having a good model of your software is the best way to communicate with other developers working on the project and with your customers. A good model is extremely important for medium and big-size projects, but it is also very useful for small ones. Even if you are working ona small one man project you will benefit from a good model because it will give you an overview that will help you code things right the first time.

UML is the diagramming language used for describing such models. You can represent your ideas in UML using different types of diagrams. Umbrello UML Modeller 2.11 supports thefollowing types:

- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- Use Case Diagram
- State Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Entity Relationship Diagram

Q1: Select and Write down the problem statement for a real time system of relevance.

Statement*:* Inventory Management Problems and Solutions

1. Lack of Inventory Visibility

If you're unable to locate or identify stocks in your inventory, shipping products on time becomes very difficult, and this can dent your business reputation. Inventory that is incomplete, difficult to find, or erroneous, is sure to hamper your bottom line. In fact, the most common reason for delayed, wrong, or partial shipments is the difficultyof locating or identifying inventory in the warehouse.

Receiving in finding the correct stock is critical for ensuring warehouse efficiency, as well as good experiences forthe customer.

Solution: Real-time inventory management system

When you implement a real-time inventory management system like Tranquil ERP, you will have all the accuratedetails regarding location data, and stock availability.

This will help in the easy location of stocks, which ultimately translates to better order fulfilment and customersatisfaction.

2. Inefficient Inventory Management Process or Software

This is probably the most common, and the biggest inventory challenges. Many businesses still try to manage their inventory with manual procedures, or outdated legacy software – which may stunt your business growth. It may not seem much of a bother to use manual, labor-intensive, or low-tech systems when you're a small, one- warehouse business but that will change when you expand. When sales volumes balloon, you need to expand inventory and add warehouses. Old and inefficient inventory management practices will be tough to scale, and

prove to be a handicap, and not give you the results you need. Manual inventory tracking procedures

involve either paperwork or tracking procedures across multiple spreadsheets and software, which can lead to data redundancy, incomplete data, and a lot of time spent on it; it also provides less security. In today's competitive age, it is essentialto know at any point in time, exactly what inventory you have; you can no longer rely on physically counting inventory annually using all your employees. Another big problem is inventory loss caused by theft, damage, spoilage, and so on; this supply chain issue requires the ability to properly pinpoint, track, and measure the problemareas. You can scale inventory management software and support complicated logistics, only when you are able to integrate it with your existing business software.

Solution: IoT-driven inventory management solution

An inventory management system that leverages IoT will empower your warehouse staff, and help them manage stocks and track them throughout their time in your warehouse efficiently, and quickly. You can streamline your inventory management with the right solution, and enhance inventory efficiency. It's not just large corporations, but even small businesses that can benefit from a centralized inventory tracking system with integrated accounting features – all of which can be found in Tranquil ERP; it is robust, intuitive, economical, flexible, and scalable.

3. Tracking Obsolete Material

In almost every business, you are likely to face this problem at some time or the other.

There will be some products or materials that remain unsold or unused, and they may become obsolete, or past theirexpiry date. These materials or products tend to accumulate over time as they are mostly ignored by inventory managers. When that product or material is needed sometime in the future, unfortunately, the unsold stock stays forgotten, and new stock is purchased; the older one may remain in the warehouse for so long that it gets damaged completely. This increases expenses, and material wastage.

Solution: Efficient stock control system

This feature is included in the Inventory management module of Tranquil ERP. You will be able to locate the deadstock and make proper use of them. Such a software solution can help inventory managers significantly in controlling stocks.

4. Identifying Incorrectly Located Materials

When there is no proper system to track products, materials, or equipment in the store, it can be cumbersome and time-consuming to find them when you have sales orders. After all, a warehouse may typically store thousands of products. This can delay sales and make customers unhappy.

Solution: Product finder

All products that you have should be tagged with RFID, barcodes, or QR codes etched with a laser.

This will help your employees identify the products that are needed. They only need to be equipped with a scanner. Once the scanner finds the required product, a lamp glows, indicating a match. This helps your pickers to quickly find the product and send it to the sales agents, saving them time, speeding up the sales cycle, and making customers happy.

5. Keeping up with Overstocks

When you purchase new materials with a few unsold products lying in your warehouse, it can affect your profitability. This situation mostly arises due to the inefficiencies of manual processes, which causes poor control of stock. Storing too much stock is as bad as storing too little, as overstocking hampers your cash flow and creates problems related to inventory, like storage, or loss.

Solution: Stock audit process

When you implement a stock audit process, inventory managers will be able to audit stocks regularly so that unused stocks are quickly identified. This boosts the efficiency of inventory greatly, enabling your company to cut costs, eliminate delays, and enhance profitability.

6. Managing Inventory Waste & Defects

Though it may seem small, it is one of the most common and repetitive inventory management problems that can cause huge losses eventually. To be able to fulfil orders in time, it is essential that you maintain optimal inventory.

Without standard procedures and untrained operators, you could end up with inventory that gets damaged

or wastedand this can not only prove to be very expensive but also lead to dissatisfied customers.

Solution: Modern inventory management software

Tranquil ERP has a robust inventory management module that enables you to manage and control your inventoryefficiently. With streamlining of procedures and processes, your employees will find it easy to perform their jobs,and you can cut costs and eliminate wastage.

7. Lack of Centralized Inventory Hub

Stocktaking becomes very challenging when you have inventories in multiple locations. Discrete stock data from various locations makes shipping complex, resulting in delays. It's one of the biggest and continual challenges faced by most businesses today.

Solution: Central inventory system

You can significantly reduce expenses and save a great deal of time by simply creating a centralized inventory hub for your inventory-related data, including stock-taking. This gives you comprehensive visibility and control of inventory and data in one single location, making stock management simple. It also becomes much easier to track the inventory that enters and leaves your business premises.

8. Changing Demand

Consumer demand is in a constant state of flux; this makes storing inventory complicated. How much to store? Toomuch, and you could end up with dead stock; too little, and you won't be able to fulfill customer demands.

Solution: Technology to Plan Inventory

What you need is robust inventory forecasting technology that takes into consideration all these factors, and helps you plan your inventory more efficiently. Tranquil ERP's inventory management module has the forecast feature that helps create and implement the optimal inventory plan. This can help you to keep up with fluctuating customer

demand.

9. Supply Chain Complexity

International supply chains are dynamic and can create roadblocks in the management and planning of your inventory. Manufacturers and distributors are impacted by unforeseen economic booms and slumps which impact raw material prices and availability. They also decide when, how, and where to ship the inventory – and this meansyou have lead times that you cannot predict, necessitating you to be much more flexible.

Solution: Robust Inventory Management application

With the right inventory management application implemented in your business, you can predict lead times as closeto accuracy as possible, and be better prepared to handle supply chain complexities.

10. Managing Warehouse Space and Efficiency

One of the most challenging tasks for any business is the efficient management of space. Warehouses need to be planned and designed with the help of inventory management platforms so that you can control when new stock is delivered, and help you make the best use of available space. If you deal in fragile or perishable products, you need to arrange specialized care and storage – for example, cold storage. You have to implement specific strategies for expensive inventory, to prevent theft and damage.

Warehouse inventory control is labor-intensive, and necessitates multiple steps like receiving the stocks, putting them away, picking inventory, packing, and finally, shipping. It is critical that all of these tasks are executed as efficiently as possible.

Solution: Warehouse Management System

Inventory management modules either have warehouse management as a feature or are integrated with warehouse management modules. This can help you automate many of the tedious tasks and bring in more efficiency in the entire warehouse management of your business.

Q2: Analyze requirement for a system and develop software Requirement Specification sheet (SRS) for suggested system.

REQUIREMENTS:

1. Identify Stakeholders: Determine the individuals or entities with an interest or involvement in the system, such as users, administrators, developers, and managers.

2. Gather Requirements: Engage in discussions, interviews, and surveys to collect information about the system's purpose, functionality, and constraints. Consider the following types of requirements:

a. Functional Requirements: Specify the system's intended behavior and functionality. For example, "The system shall allow users to create and edit profiles."

b. Non-functional Requirements: Define the qualities or characteristics of the system. This may include performance, security, usability, or reliability requirements. For example, "The system should respond to user actions within 2 seconds."

c. User Requirements: Capture the needs and expectations of the system's end-users. For example, "The system should have an intuitive and user-friendly interface."

d. System Requirements: Consider technical aspects and constraints, including hardware, software, and compatibility requirements. For example, "The system should be compatible with Windows 10 and macOS 11."

3. Analyze and Prioritize Requirements: Review and organize the gathered requirements, ensuring they are complete, consistent, and unambiguous. Identify dependencies and prioritize the requirements based on their importance and impact on the system.

4. Define System Architecture: Outline the high-level structure and components of the system. This may include diagrams, such as a system architecture diagram or data flow diagram, to illustrate the interactions between different system modules.

5. Create SRS Document: Based on the gathered requirements and analysis, develop the Software Requirement Specification (SRS) document. The SRS should include the following sections:

a. Introduction: Provide an overview of the system, its purpose, and scope.

b. System Overview: Describe the system's major components, their interactions, and any external interfaces.

c. Functional Requirements: Document the functional requirements in detail, specifying inputs, outputs, and expected behaviors.

d. Non-functional Requirements: Include the non-functional requirements, such as performance, security, and usability, along with any specific metrics or constraints.

e. User Requirements: Present the user's needs and expectations in terms of usability, accessibility, and user experience.

f. System Requirements: Outline the technical requirements, including hardware, software, compatibility, and performance criteria.

g. Constraints: Specify any limitations or constraints that may impact the system's design or implementation.

h. Dependencies: Identify any dependencies or interactions with external systems or components.

i. Assumptions and Dependencies: Document any assumptions made during the requirement gatheringprocess and dependencies on other systems or components.

j. Glossary: Provide a glossary of terms used throughout the document to ensure a common understanding.

k. Appendices: Include any additional supporting information, such as diagrams, mockups, or supplementary documents.

6. Review and Validate: Conduct a thorough review of the SRS document, involving stakeholders andsubject matter experts, to validate and refine the requirements.

Once the SRS document is complete and approved by all relevant stakeholders, it serves as a reference for the system's design, development, and testing phases. It helps ensure that the system meets the specified requirements and provides a basis for effective project management.

SOFTWARE REQUIREMENT SHEET

1. Introduction
   - Purpose
   - Scope
   - Document Conventions
   - Intended Audience
   - References

2. System Overview
   - System Description
   - System Components
   - External Interfaces
   - System Architecture

3. Functional Requirements
   - User Registration
   - Create and Edit Profiles

4. Non-functional Requirements
   - Performance
   - Security

5. User Requirements
   - Intuitive and User-Friendly

Multilingual Support

6. System Requirements
   Scalability
   Cross-platform compatibility

7. Constraints
   Compliance with Regulatory Standards
   Integration with Existing Systems

8. Dependencies
   Payment Gateway Integration
   Email Service Provider Integration

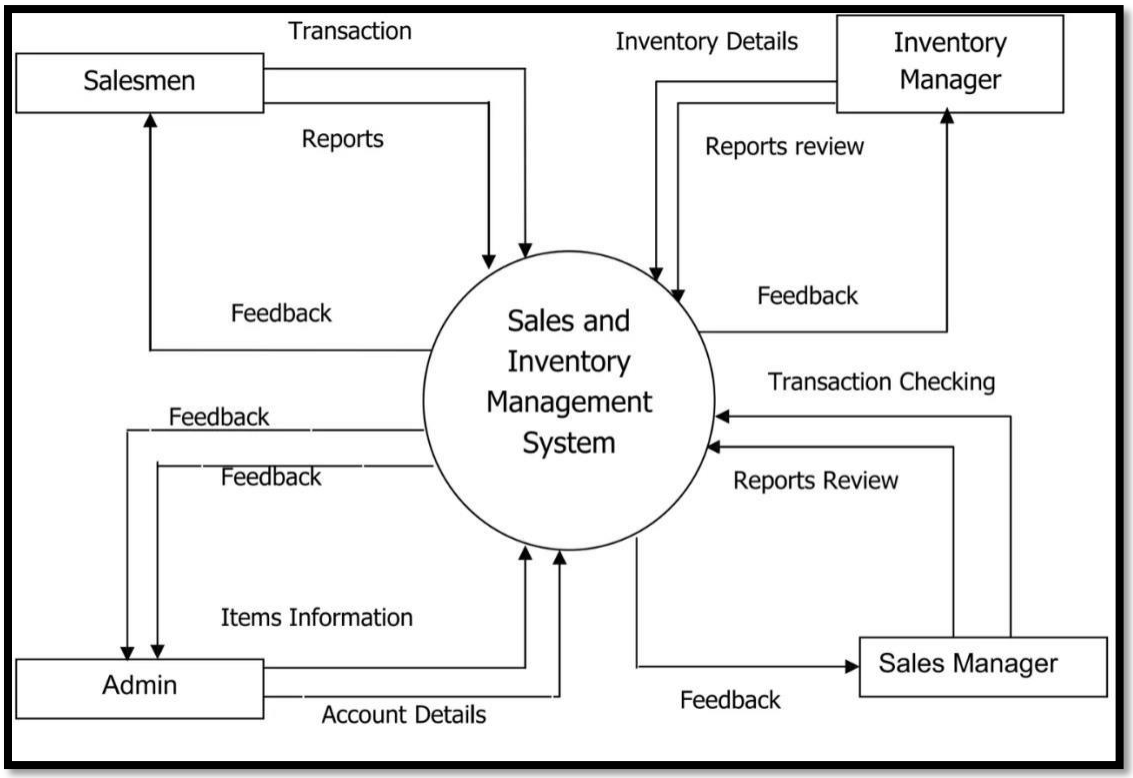
9. Assumptions and Dependencies
   Assumptions
      User Authentication
      Adequate System Resources
   Dependencies
      External API Integration
      Cloud Storage Provider

10. Glossary

11. Appendix

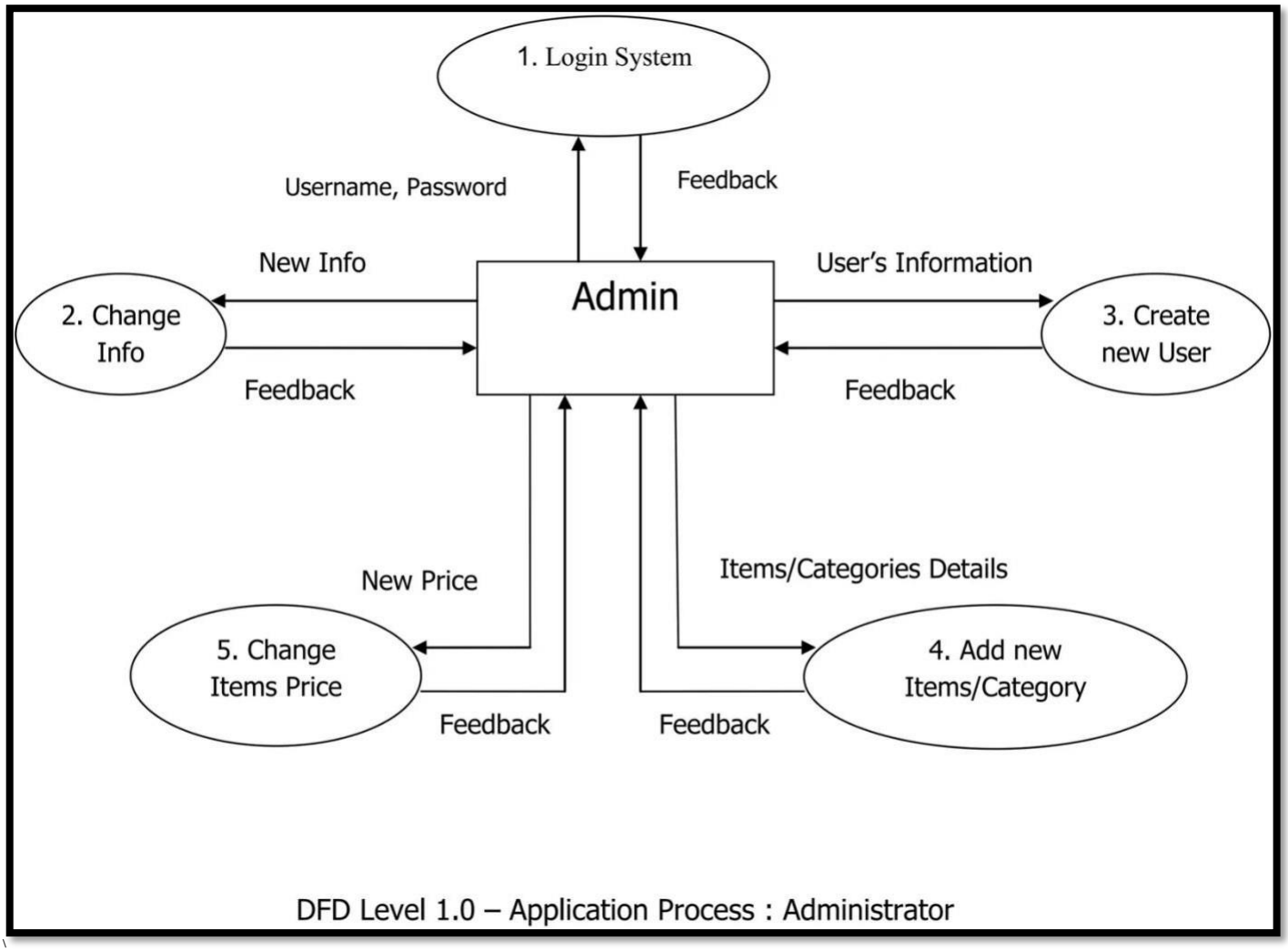
Q3: To create the function oriented diagram: Data Flow Diagram (DFD)

Level 0 DFD



At this level, the system has to show with more details of processing. The processes that are necessary to be carried out are book delivery and search by topic. Let us follow this up with a level 1 DFD of the same LMS.

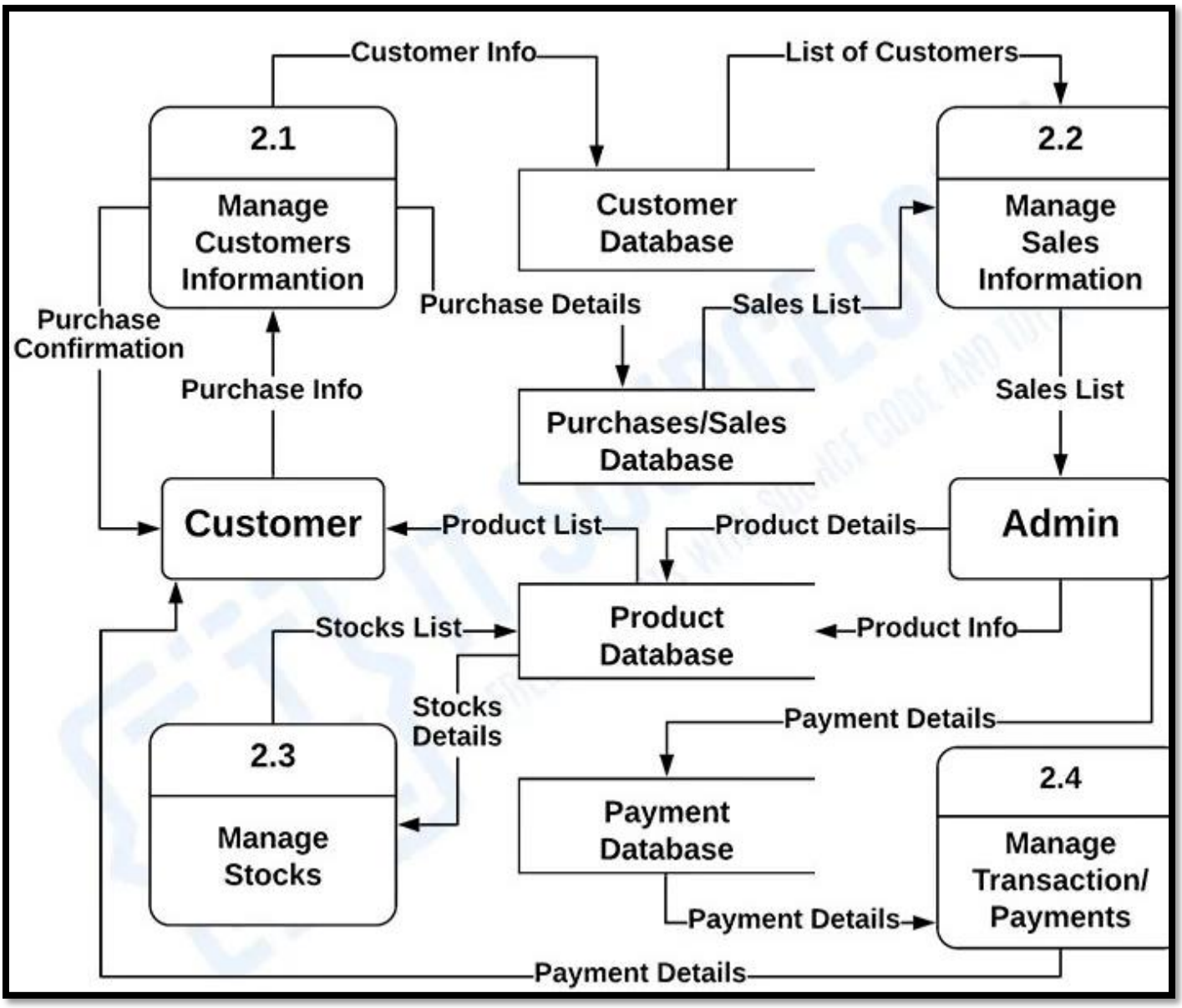
Level 1 DFD



DFD Level 1.0 – Application Process : Administrator

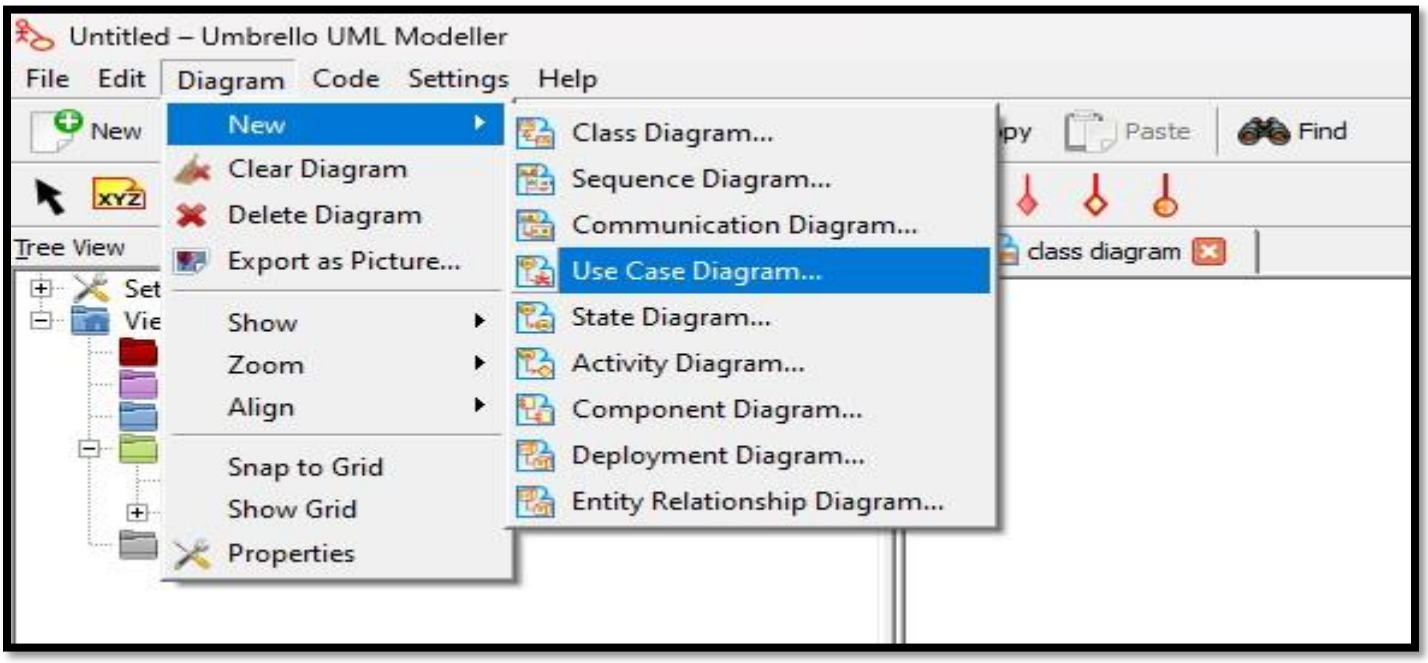
At this level, the system has to show or exposed with more details of processing.The processes that are important to be carried out are:

Level 2 DFD



So as we go into more and more details of the Data Flow Diagram we get to know the flow within each process of function hence getting able to know the inner details of the system which we choose.

Q4: To perform the user's view analysis for the suggested system: Use case diagram.



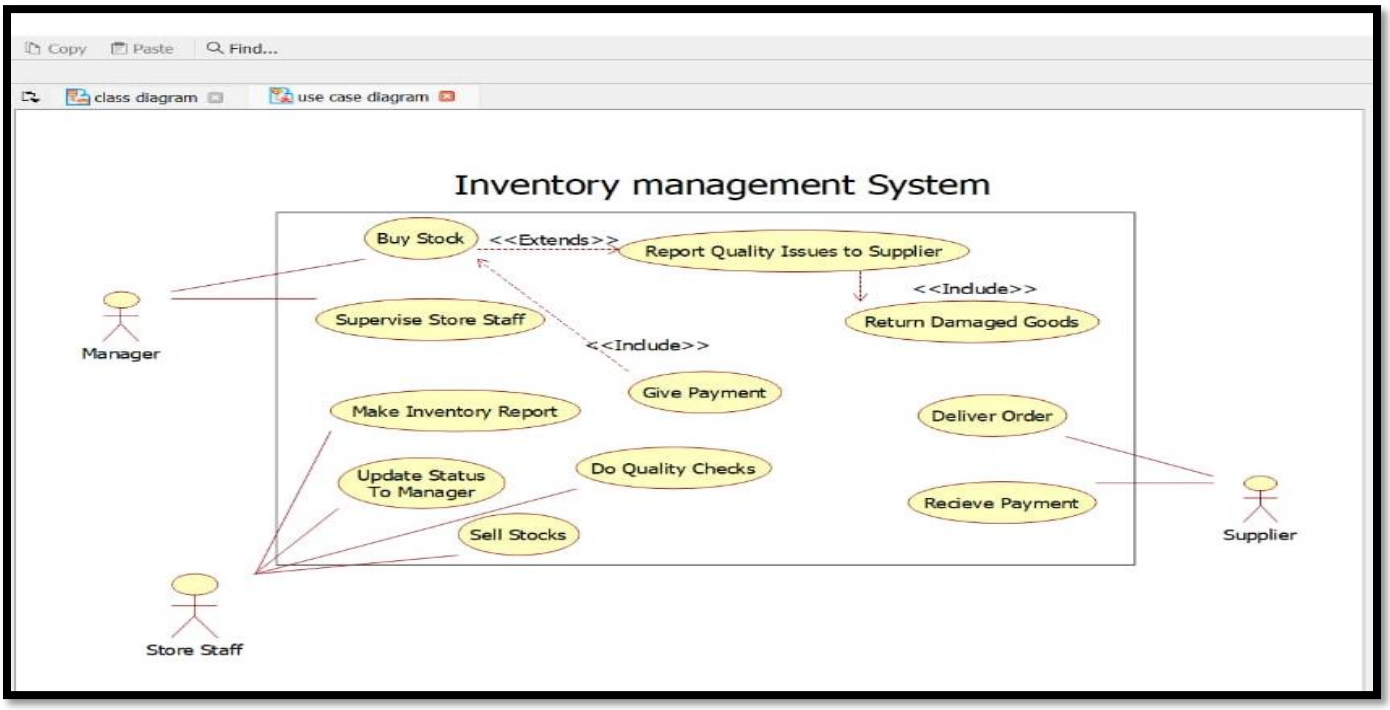
To create a new Umbrello Use case diagram canvas, follow these steps:

- Launch Umbrello: Open the Umbrello software on your computer. Ensure that you have installed Umbrello and it is accessible from your application menu or desktop

- Create a new project: Click on "File" in the menu bar and select "New Project" or use the shortcut Ctrl+N. Choose a location to save the project and give it a name

- Select use case diagram: Once the project is created, go to "Diagrams" in the menu bar and choose "New Diagram" or use the shortcut Ctrl+Shift+D. In the dialog box that appears, select "Use Case Diagram" as the diagram type.

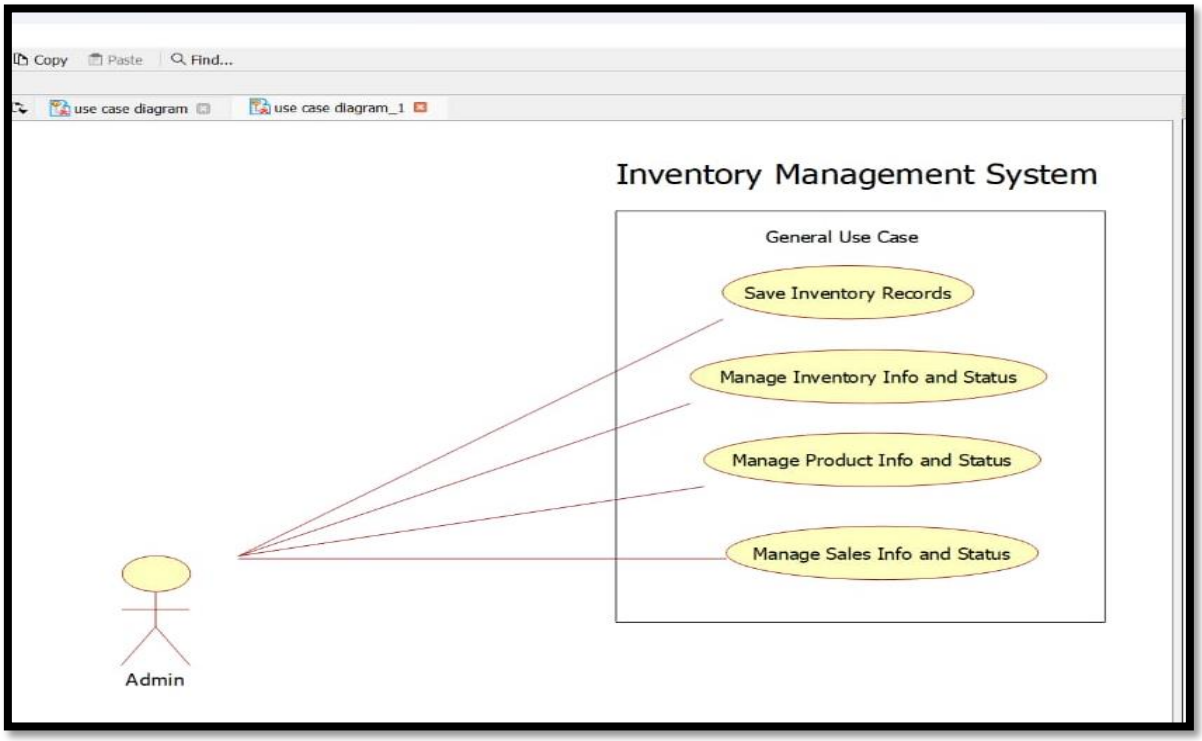
DIAGRAM-1



In above use case diagram:

- Primary actors Manager and Store Staff interact with the use cases like Buy Stock, Supervise Store Staff, Sell Stocks, Make Inventory Report etc. They are primary actors because they take assistance from the system
- Secondary actor like Supplier also interact with the system but this time being a secondary actor they provide assistance to the system like use cases and the primary actor so that the interaction does not stop and keeps going on

DIAGRAM-2



In this use case diagram:

Actors:

"Inventory Manager": Represents a user or role responsible for managing the inventory. "Warehouse Staff": Represents a user or role responsible for managing the warehouse operations. Use Cases:

"Manage Inventory": Represents the use case where the Inventory Manager and Warehouse Staff interact with the system to perform tasks related to inventory management, such as adding items, updating item information, and tracking inventory levels.

"Generate Reports": Represents the use case where the Inventory Manager generates reports basedon inventory data, such as inventory levels, stock status, or sales reports.

"Place Order": Represents the use case where the Warehouse Staff places an order for items orsupplies.

"Receive Order": Represents the use case where the Warehouse Staff receives an order and updatesthe inventory accordingly.

Relationships:

The "Inventory Manager" actor is associated with the "Manage Inventory" use case, indicating their involvement in managing the inventory.

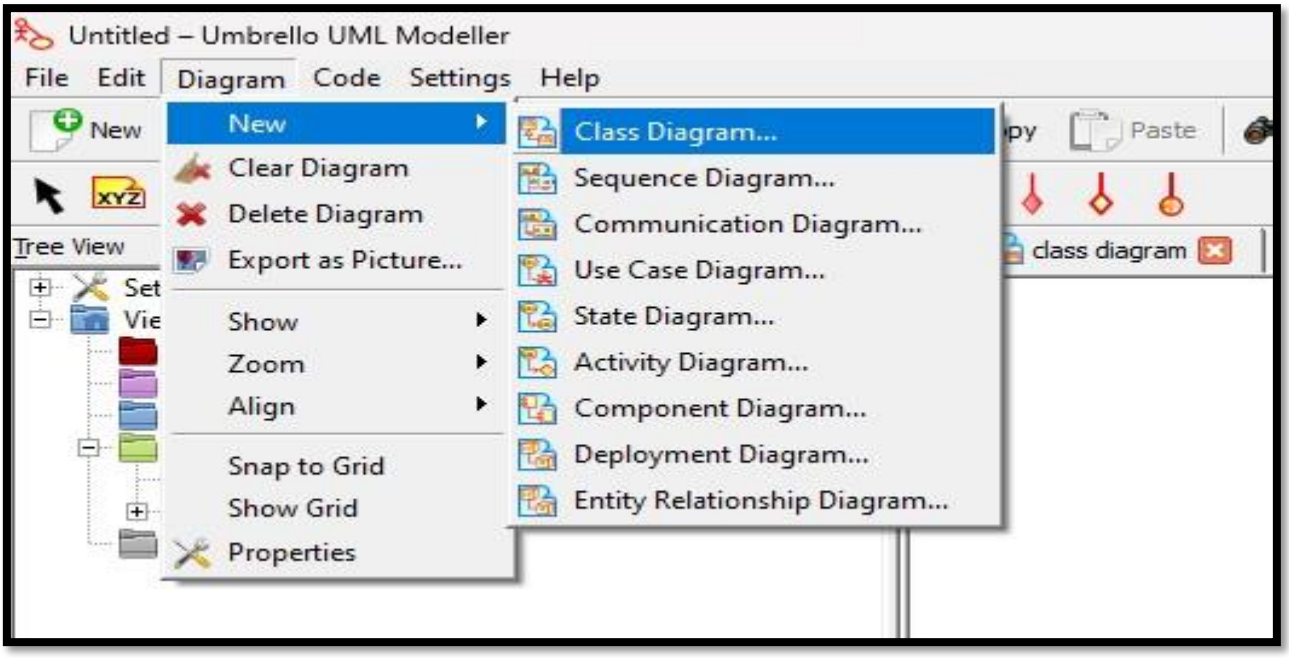
The "Inventory Manager" actor is also associated with the "Generate Reports" use case, representing their ability to generate inventory-related reports.

The "Warehouse Staff" actor is associated with the "Manage Inventory" use case, indicating their involvement in performing tasks related to inventory management.

The "Warehouse Staff" actor is associated with the "Place Order" use case, indicating theirresponsibility for placing orders.

The "Warehouse Staff" actor is associated with the "Receive Order" use case, indicating their rolein receiving and updating the inventory after an order is received.

Q5: To Draw the structural view diagram for the system: Class diagram

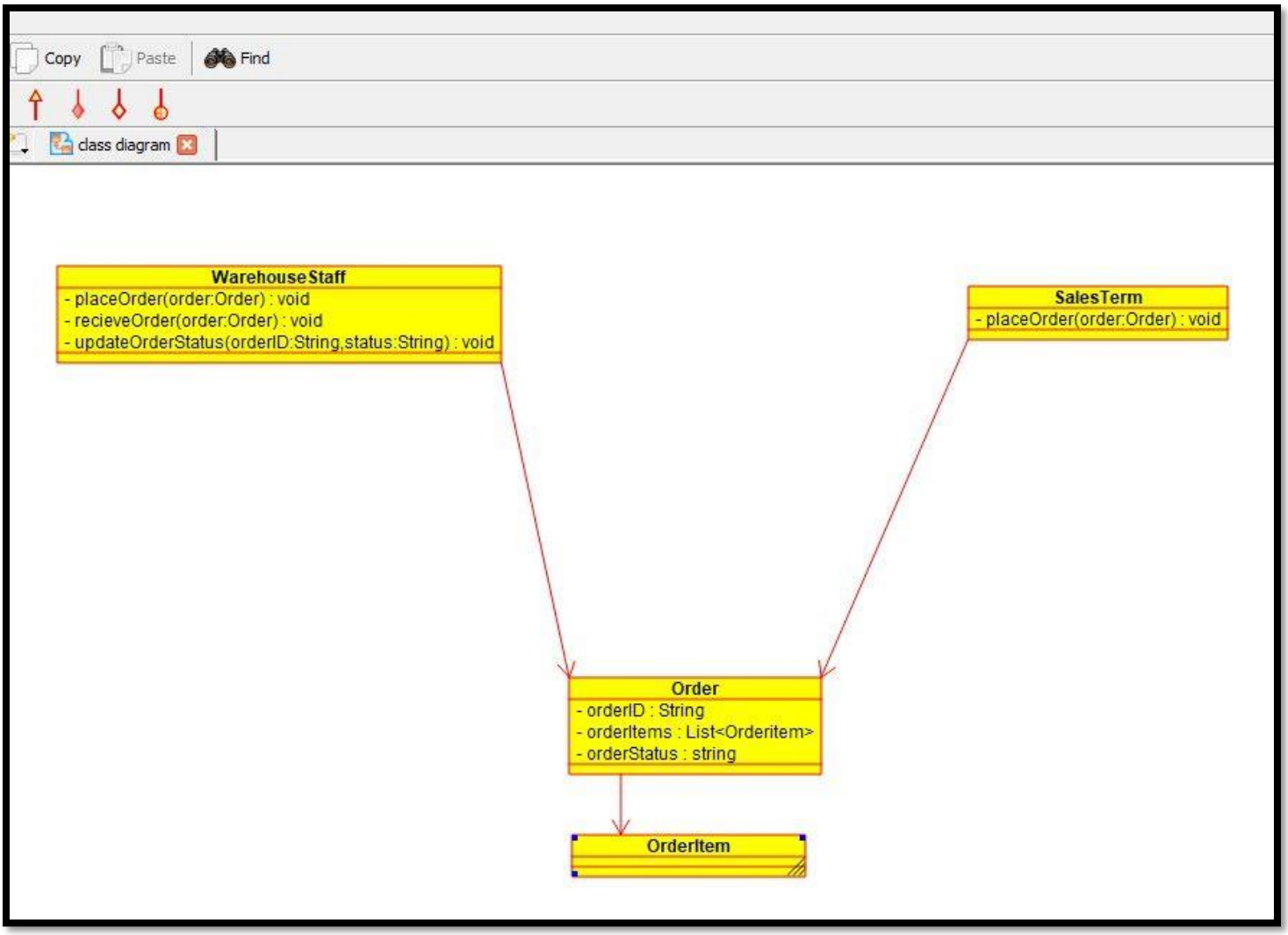


To create a new Umbrello Class diagram canvas, follow these steps:

- Launch Umbrello: Open the Umbrello software on your computer. Ensure that you have installed Umbrello and it is accessible from your application menu or desktop

- Create a new project: Click on “File” in the menu bar and select “New Project” or use the shortcut Ctrl+N. Choose a location to save the project and give it a name

- Select class diagram: Once the project is created, go to “Diagrams” in the menu bar and choose “New Diagram” or use the shortcut Ctrl+Shift+D. In the dialog box that appears, select “Class Diagram” as the diagram type.

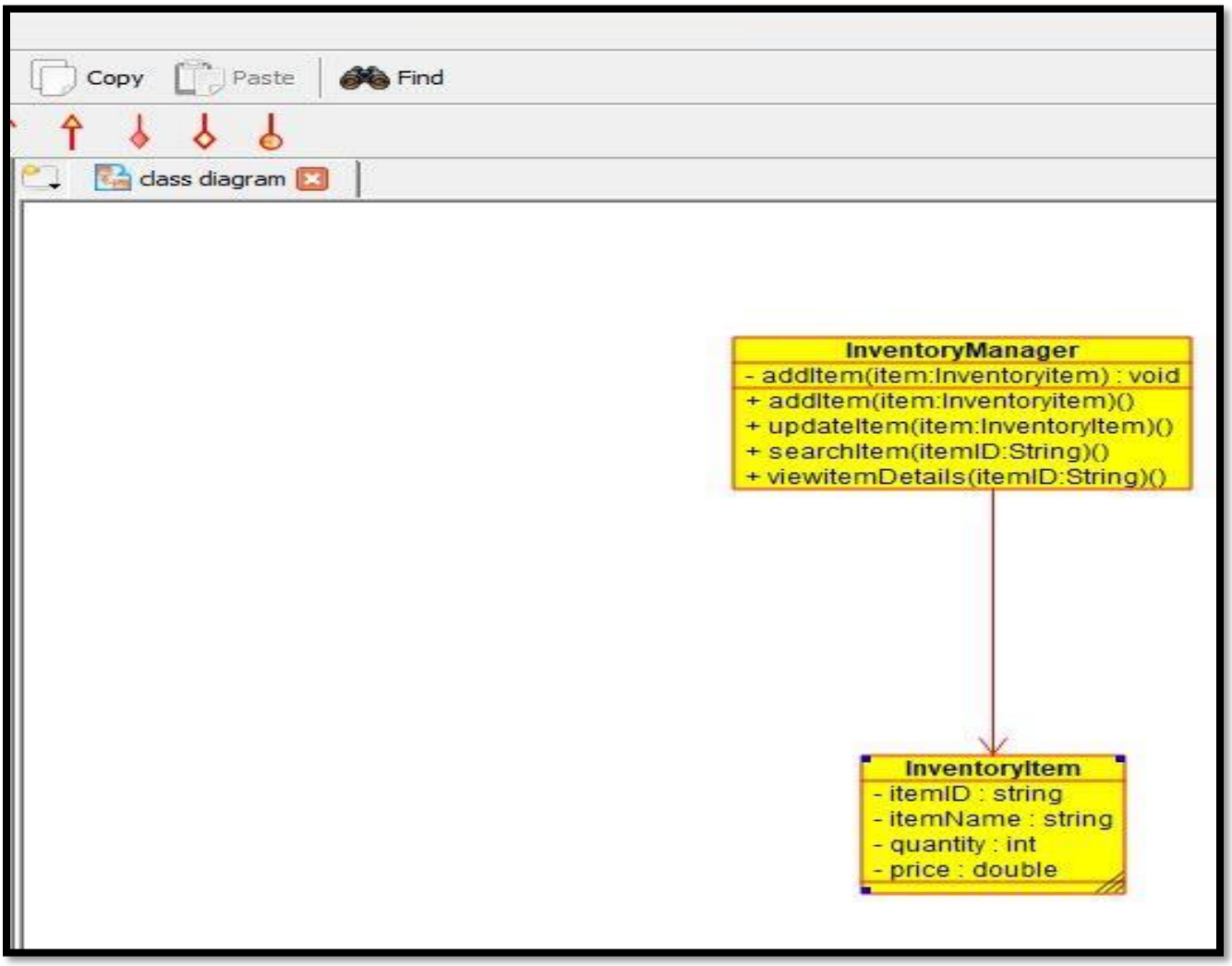


In this Class Diagram, the structure of an inventory management system is represented. Here's anexplanation of what's happening:

Classes:

- "InventoryManager": Represents the inventory manager in the system. It has attributes such as "inventoryRecords" (a list of inventory items) and methods like "addItem," "updateItem," "searchItem," and "viewItemDetails."
- "WarehouseStaff": Represents the staff responsible for warehouse operations. It has methods for "placeOrder" (to place an order), "receiveOrder" (to receive an order), and "updateOrderStatus" (to update the status of an order).
- "SalesTeam": Represents the team responsible for sales. It has a method for "placeOrder" (toplace an order).

- "InventoryItem": Represents an individual item in the inventory. It has attributes such as "itemID," "itemName," "quantity," and "price."
- "Order": Represents an order in the system. It has attributes such as "orderID," "orderItems" (a list of order items), and "orderStatus."
- "OrderItem": Represents an item within an order. It has attributes such as "itemID,""itemName," and "quantity."

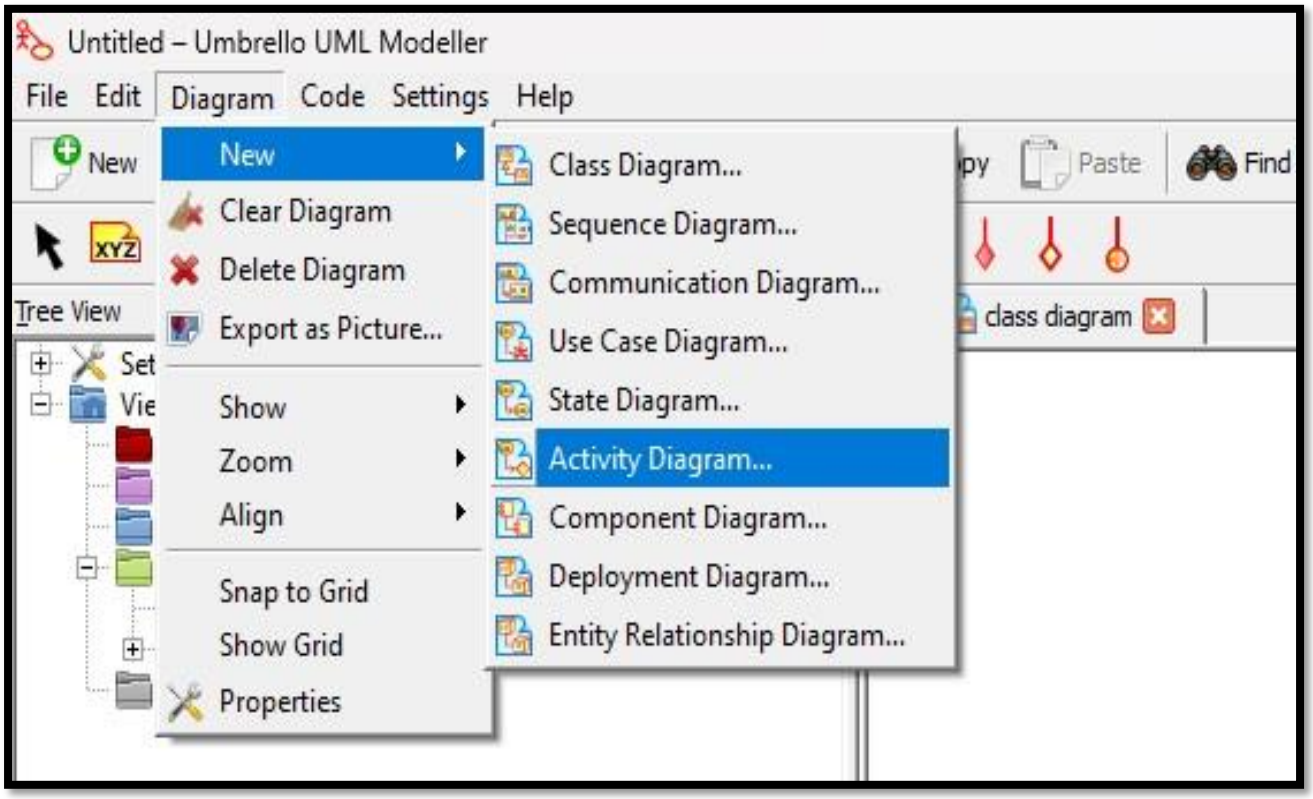


Relationships:

- "InventoryManager" is associated with "InventoryItem" since the manager can perform actions on inventory items.
- "WarehouseStaff" and "SalesTeam" are associated with "Order" since they can place orders.
- "WarehouseStaff" is also associated with "Order" for receiving and updating the status of orders.
- "Order" is associated with "OrderItem" since an order can contain multiple items.

Overall, this Class Diagram illustrates the structural relationships between the classes involved in the inventory management system, including the roles and functionalities of the inventory manager, warehouse staff, sales team, inventory items, orders, and order items.

Q6: To Draw the behavioral view diagram: State-chart diagram or Activity Diagram



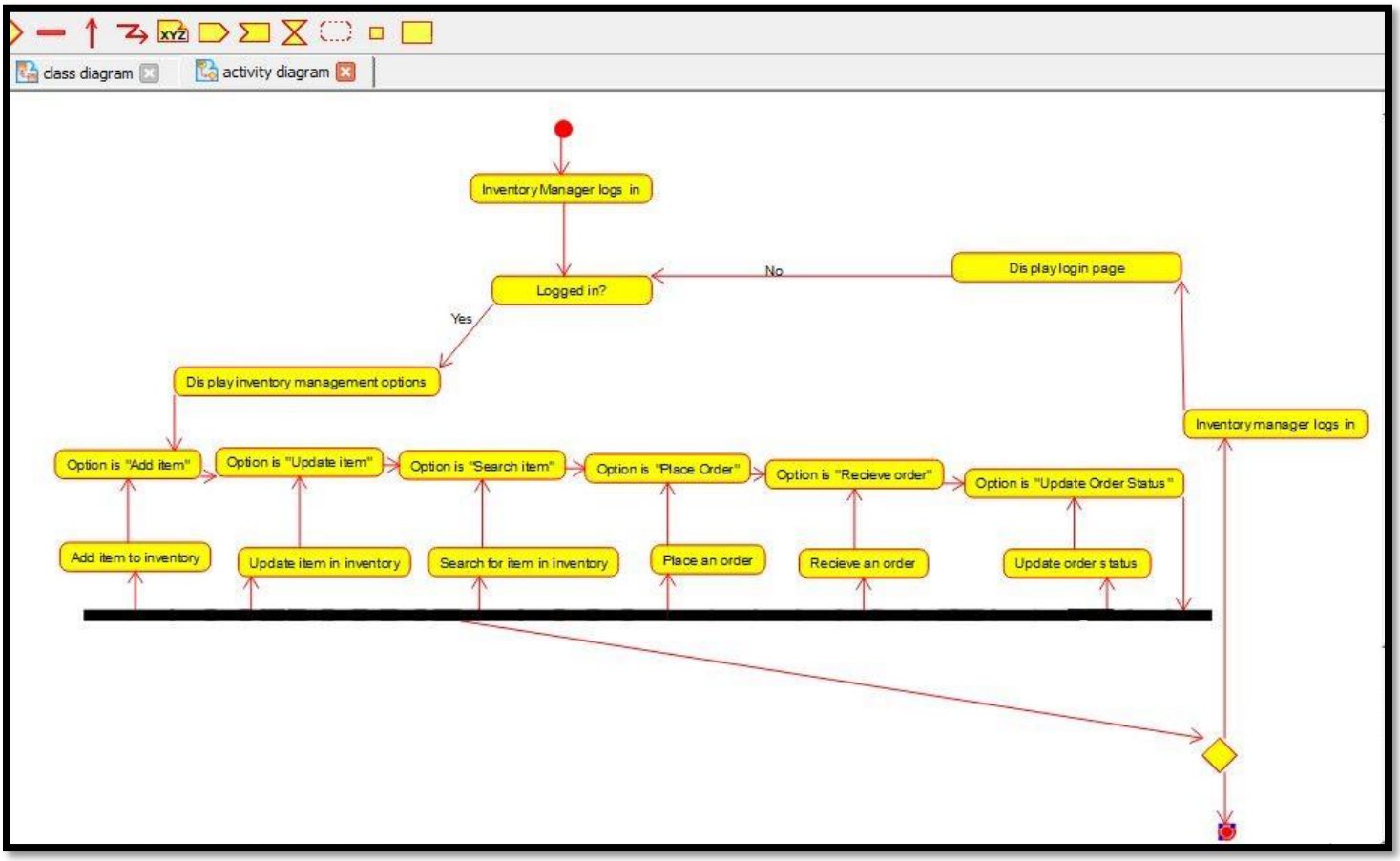
To create a new Umbrello Activity diagram canvas, follow these steps:

- Launch Umbrello: Open the Umbrello software on your computer. Ensure that you have installed Umbrello and it is accessible from your application menu or desktop

- Create a new project: Click on "File" in the menu bar and select "New Project" or use the shortcut Ctrl+N. Choose a location to save the project and give it a name

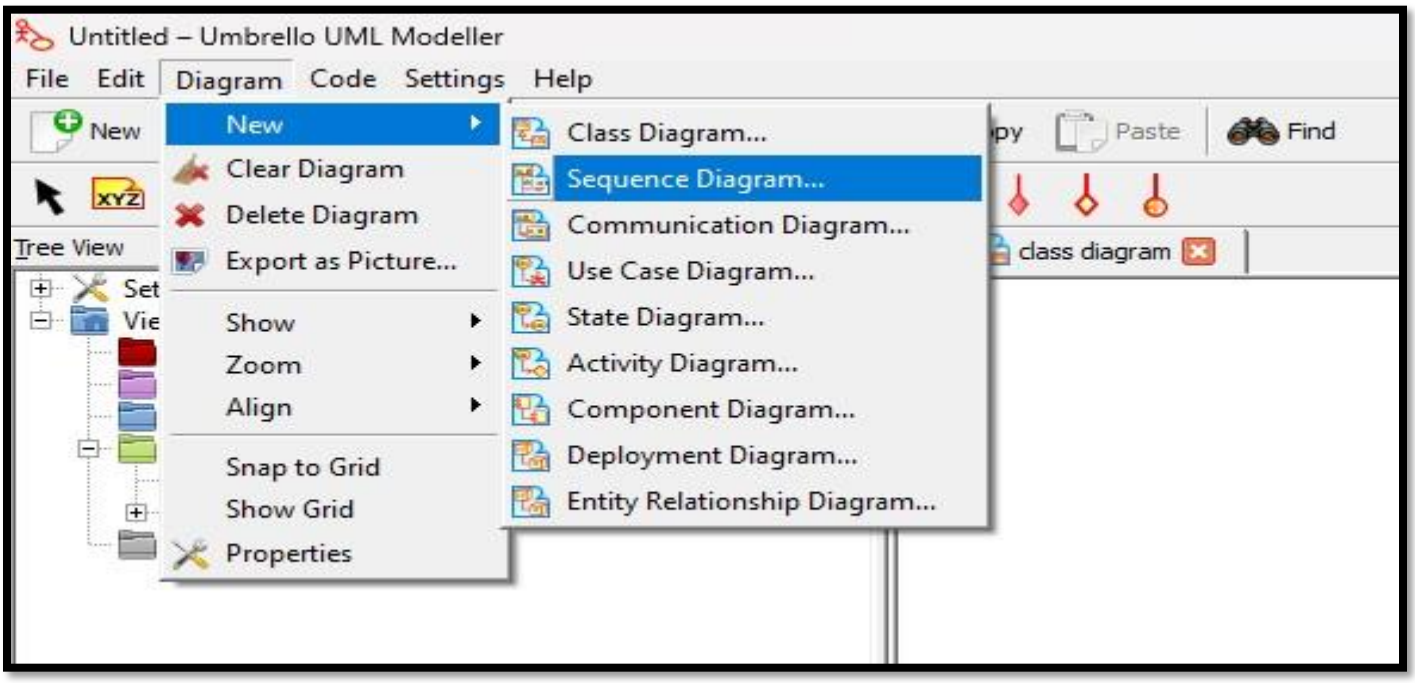
- Select Activity diagram: Once the project is created, go to "Diagrams" in the menu bar and choose "New Diagram" or use the shortcut Ctrl+Shift+D. In the dialog box that appears, select "Activity Diagram" as the diagram type.

This Activity Diagram represents:



- The process begins with the start node, and the Inventory Manager logs in.
- If the Inventory Manager is logged in, the system displays inventory management options.
- Depending on the chosen option, the system performs specific activities such as adding an item, updating an item, searching for an item, viewing item details, placing an order, receiving an order, or updating the order status.
- If the Inventory Manager is not logged in, the system displays the login page, and theInventory Manager logs in.
- Finally, the process ends with the stop node.

Q7: to perform the behavioral view diagram for the suggested system: Sequence Diagram

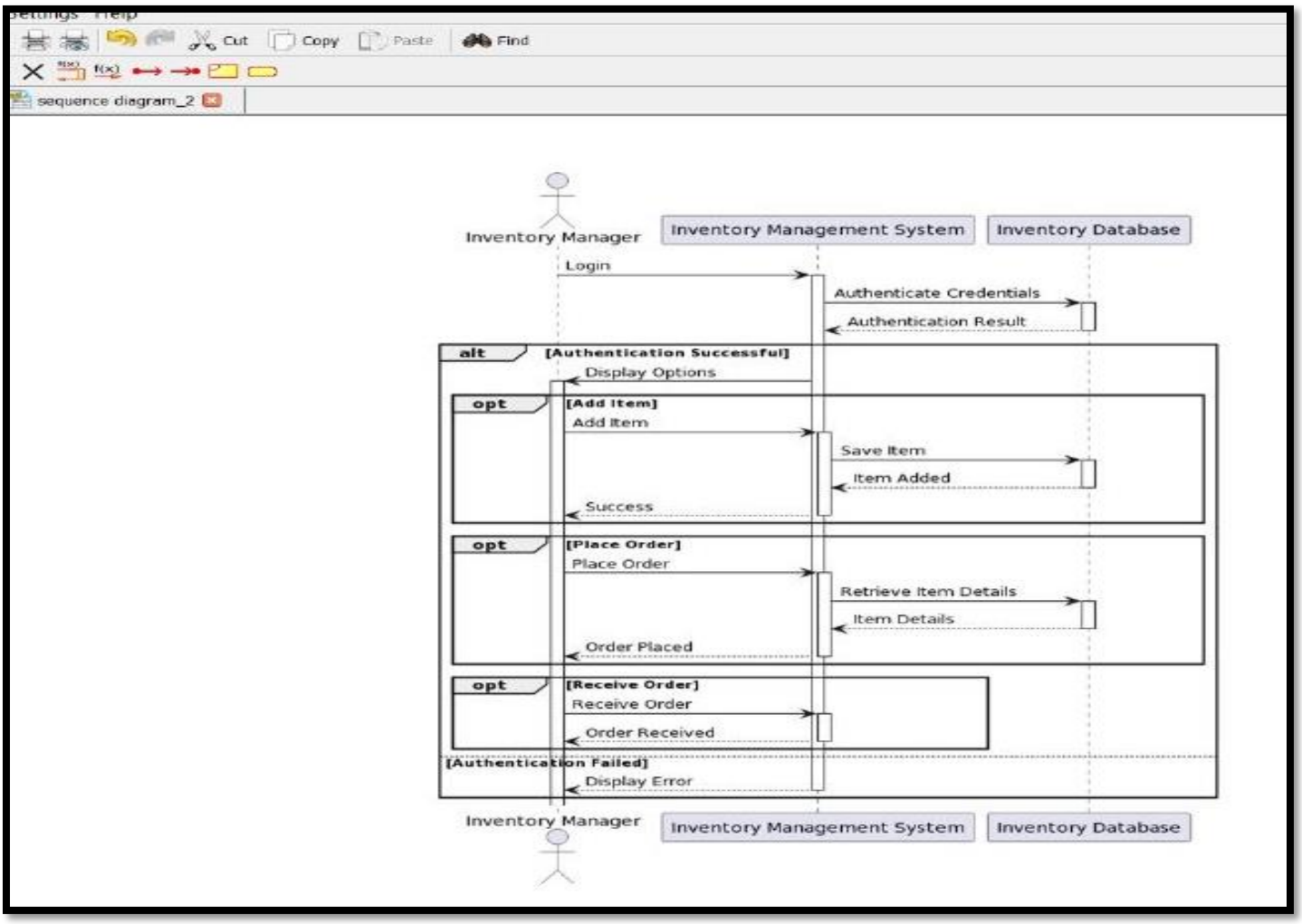


To create a new Umbrello Sequence diagram canvas, follow these steps:

- Launch Umbrello: Open the Umbrello software on your computer. Ensure that you have installed Umbrello and it is accessible from your application menu or desktop

- Create a new project: Click on "File" in the menu bar and select "New Project" or use the shortcut Ctrl+N. Choose a location to save the project and give it a name

- Select Sequence diagram: Once the project is created, go to "Diagrams" in the menu bar and choose "New Diagram" or use the shortcut Ctrl+Shift+D. In the dialog box that appears, select "Sequence Diagram" as the diagram type.



In the provided sequence diagram:

1. The Inventory Manager actor initiates the login process by sending a Login message to the Inventory Management System.
2. The Inventory Management System receives the Login message and activates itself to handle the request.
3. The system communicates with the Inventory Database by sending an Authenticate Credentials message.
4. The Inventory Database activates and processes the request, then sends back the authentication result to the system (Authentication Result message).
5. Based on the authentication result, the system follows two different paths:

a. If the authentication is successful:

- The system activates the Inventory Manager and sends a Display Options message to show available options to the manager.
- The manager can choose to perform actions such as Add Item, Place Order, or

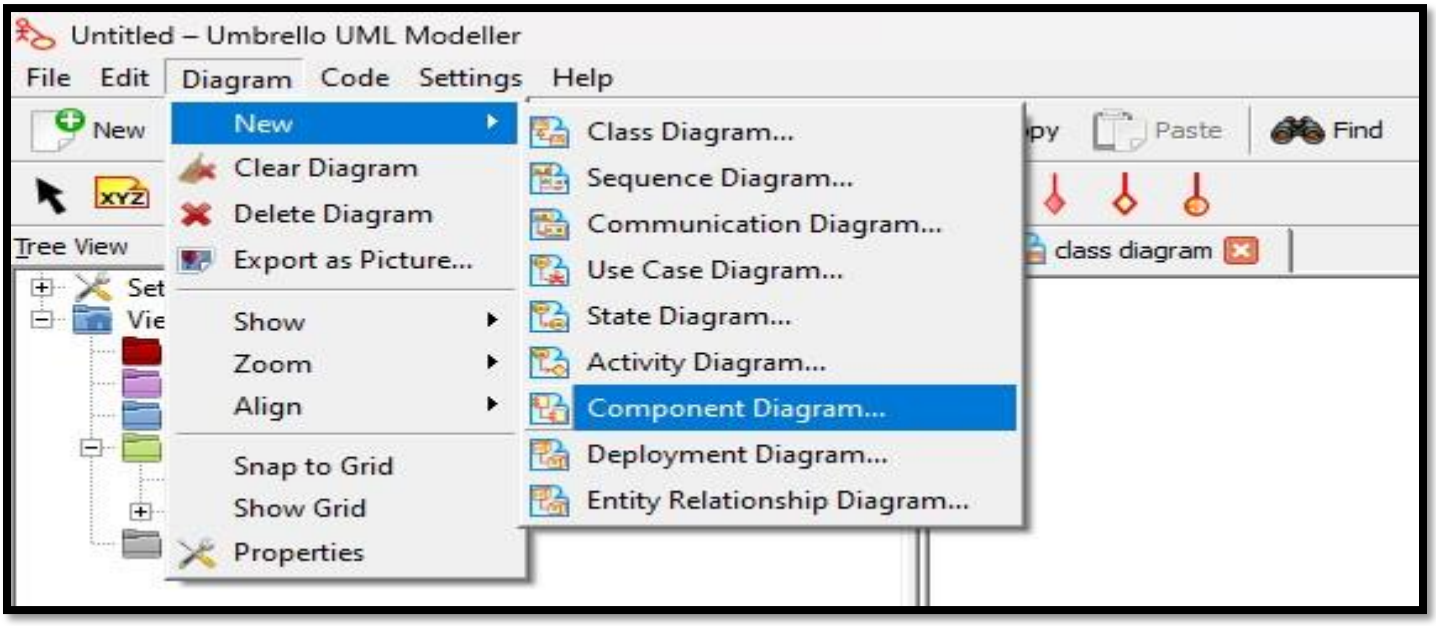
Receive Order.

- For each chosen action, the manager sends a corresponding message to the system, and the system activates the necessary components to handle the request.
- In each case, the system communicates with the Inventory Database to perform operations such as saving an item, retrieving item details, or confirming order receipt.
- Success messages (Success or Order Placed or Order Received) are sent back from the system to the manager.

b. If the authentication fails:

- The system sends an Display Error message to the Inventory Manager to show an error message indicating the failed authentication.

Q8: Draw the Component diagram

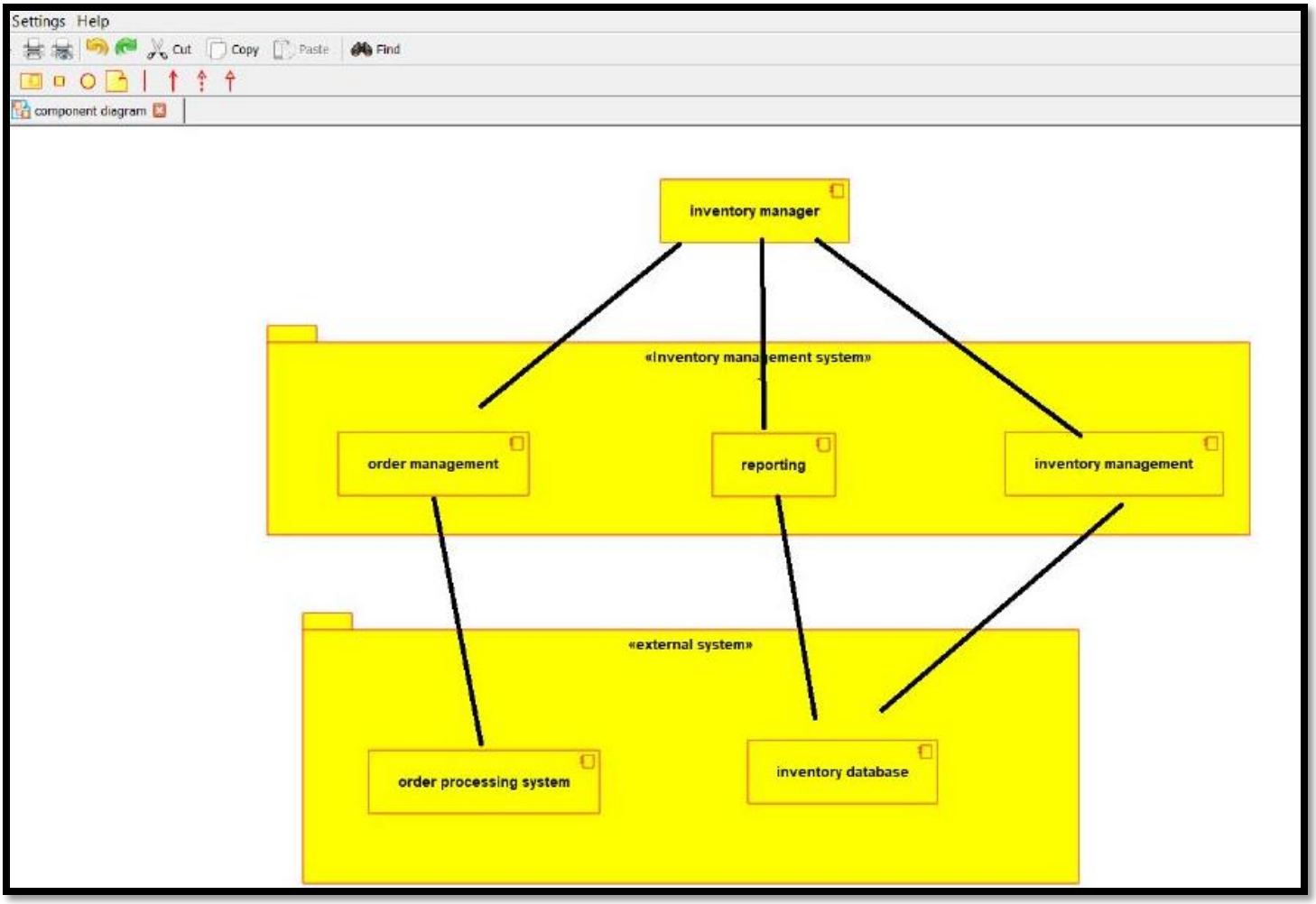


To create a new Umbrello Component diagram canvas, follow these steps:

- Launch Umbrello: Open the Umbrello software on your computer. Ensure that you have installed Umbrello and it is accessible from your application menu or desktop

- Create a new project: Click on “File” in the menu bar and select “New Project” or use the shortcut Ctrl+N. Choose a location to save the project and give it a name

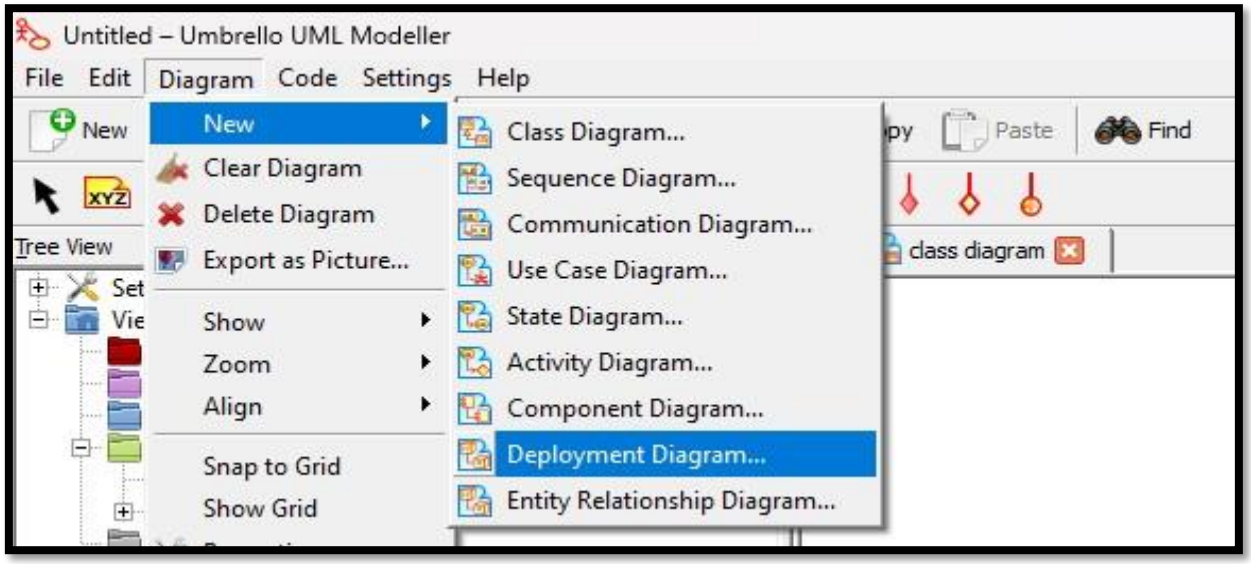
- Select Component diagram: Once the project is created, go to “Diagrams” in the menu bar and choose “New Diagram” or use the shortcut Ctrl+Shift+D. In the dialog box that appears, select “Component Diagram” as the diagram type.



In this component diagram:

- The inventory management system is divided into three main components: Inventory Management, Order Management, and Reporting.
- The Inventory Management component handles tasks related to managing the inventory, such as adding, updating, and searching items.
- The Order Management component is responsible for managing the order processing system, including placing orders, receiving orders, and updating order status.
- The Reporting component generates various reports related to inventory, sales, and reorder requirements.
- The external systems include the Inventory Database and the Order Processing System. The
- Inventory Database stores the inventory-related data and is accessed by both the Inventory Management and Reporting components.
- The Order Processing System handles order-related operations by Management.

Q9: Draw the Deployment diagram



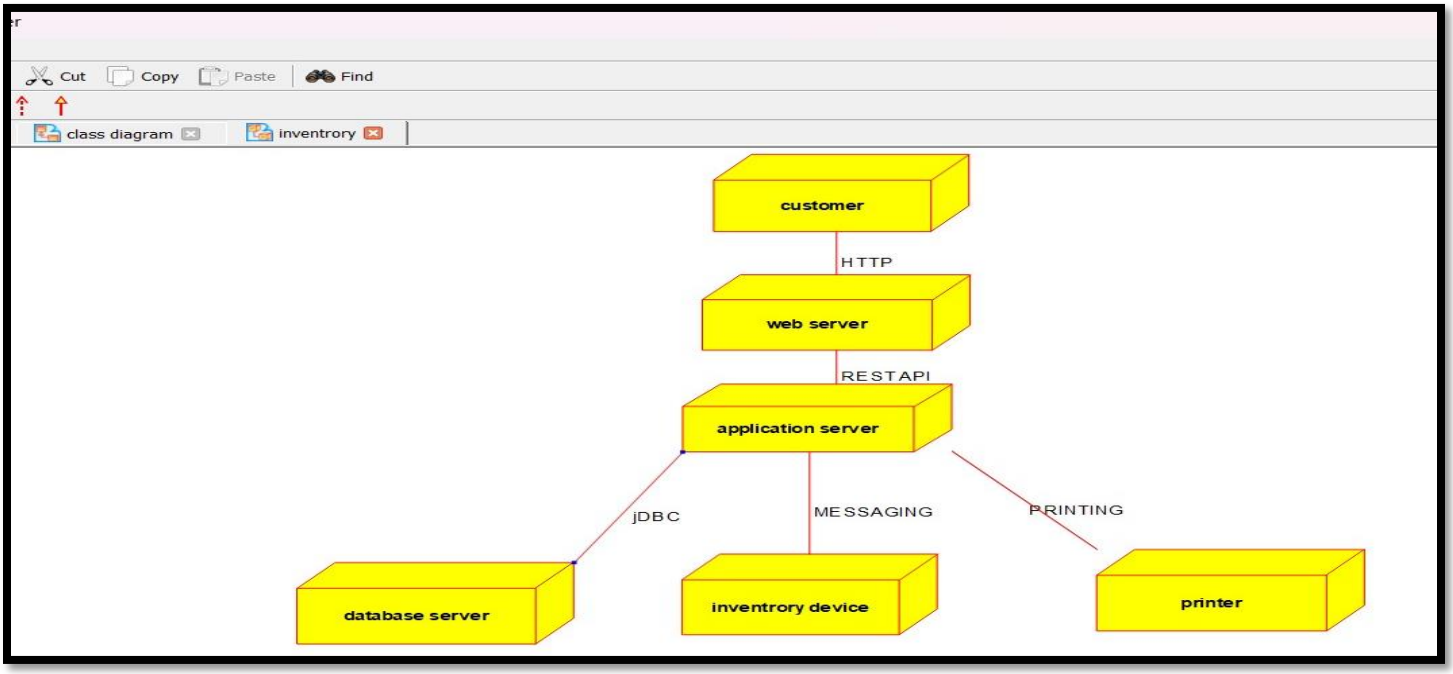
To create a new Umbrello Deployment diagram canvas, follow these steps:

- Launch Umbrello: Open the Umbrello software on your computer. Ensure that you have installed Umbrello and it is accessible from your application menu or desktop

- Create a new project: Click on “File” in the menu bar and select “New Project” or use the shortcut Ctrl+N. Choose a location to save the project and give it a name

- Select Deployment diagram: Once the project is created, go to “Diagrams” in the menu bar and choose “New Diagram” or use the shortcut Ctrl+Shift+D. In the dialog box that appears, select “Deployment Diagram” as the diagram type.

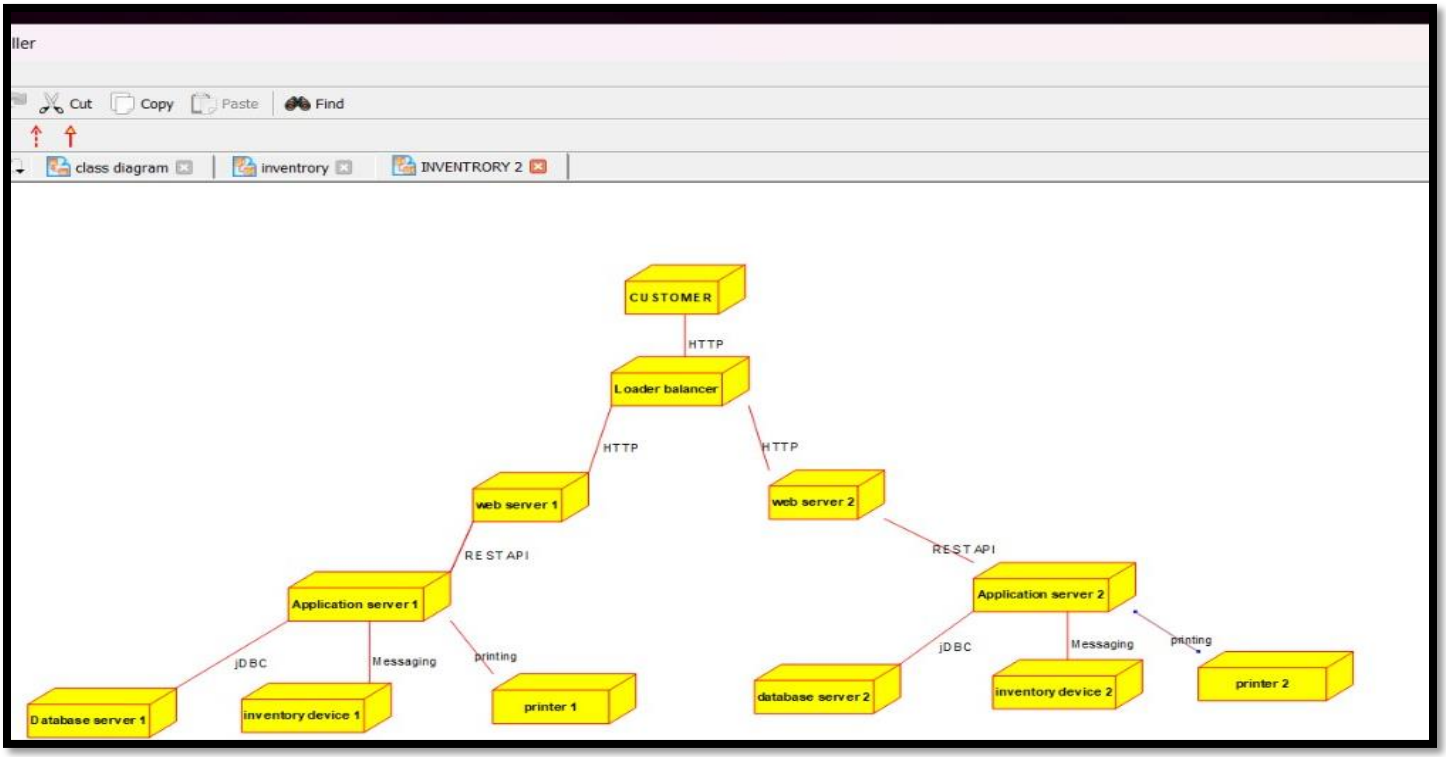
Diagram 1



In this deployment diagram:

- The inventory management system consists of several nodes: Customer, Web Server, Application Server, Database Server, Inventory Device, and Printer.
- The relationships between the nodes depict the communication channels or protocols used forinteraction, such as HTTP between the Customer and Web Server, REST API between the Web Server and Application Server, JDBC for database access between the Application Server and Database Server, messaging between the Application Server and Inventory devices
- This diagram provides an overview of the deployment architecture of the inventorymanagement system, showcasing the interactions between different components and devices.

Diagram 2



In this more detailed deployment diagram:

- The inventory management system is deployed across multiple nodes, including Customer, Load Balancer, Web Server 1, Web Server 2, Application Server 1, Application Server 2, Database Server 1, Database Server 2, Inventory Device 1, Inventory Device 2, Printer 1, and Printer 2.

- The relationships between the nodes represent the communication channels or protocols used for interaction, such as HTTP between the Customer and Load Balancer, HTTP between the Load Balancer and Web Servers, REST API between the Web Servers and Application Servers, JDBC for database access between the Application Servers and Database Servers, messaging between the Application Servers and Inventory Devices, and printing operations between the Application Servers and Printers.

Q10: Perform Measurement of complexity with Halstead Metrics for chosen system.

Halstead are based on -

Number of unique or distinct operators, n = number of unique or distinctoperands.

N1 = total number of occurrences of operators. N2 = total number of occurrences of operands.

Halstead measurements think a program as an arrangement of operators andtheir related operands.

Aside from these amounts there are others as well.

n1*= number of potential operators.n2*=number of potential operands.

Halstead Metrics are as follows:

Measure n (Length of the program)

Is the sum of the total number of the operators and operands in the program.

N=N1+N2

Measure n (Vocabulary of the program)

Is the sum of the number of unique operators and operands.

n=n1+n2

Measure V (Volume of the program)

Is the size of any implementation of any algorithm.

V=(N1+N2) log2 (n1+n2)

V=N log2 (n)

Measure D (Difficulty of the program)

Is proportional to a number of unique operators and total usage of operands.

D=(2/n1) * (n2/N2)

Measure V * (Potential or minimal Volume V *)

V*=(2+n2) log2 (2+N2)

Measure L (implementation Level)

L=V*/V

Measure L '(Program Level Estimator L ') L'=(2/nl) * (n2/N2)

Measure I (Intelligent Content)

I=L'V

Measure E – Effort required implementing or understanding theprogram is directly proportional to difficulty and volume.

E=D*V

Measure B – Number of bugs expected in the program.

Is proportional to the effort

B=E0.667. /3000

Measure T – Estimated time is taken to write the program.

Is proportional to the effort.

T=E/S [S=18]

=18]

Operators	Occurrences	Operands	Occurrences
Int	5	Search	1
()	5	a	8
,	4	b	3
[]	7	i	8
If	3	j	7
<	3	save	3
;	12	Im1	3
For	2	2	2
=	6	1	3
-	1	0	1
<=	2	-	-
++	2	-	-
Return	2	-	-
{}	3	-	-
n1=15	N1=57	n2=12	N2=39

Here N1=57 and N2=39. The program length N=N1+N2=96 Vocabulary of the program n=n1+n2=27

Volume V= N* $\log_2$N=96 a log2 27=420 bits.

The estimate program length N of the program

= 15 $\log_2$ 15+10 $\log_2$)10

=15* 3.81+ 10* 4.32

=54.45+33.2=87.65

The potential Volume V*=5$\log_2$ 5=11.6

Since L=V*/V

= 12.6/420

= 0.034

D=I/L

= 1/0.03

= 3.33

Estimated Program Level

L= 2/n1 x n2/N2= 2/15 x 10/38

= 0.041

= 420/0.034

= 12352.94

Therefore, 12352 elementary mental discrimination is required to construct the program.

$$T = E/B$$

$$= 12352/18$$

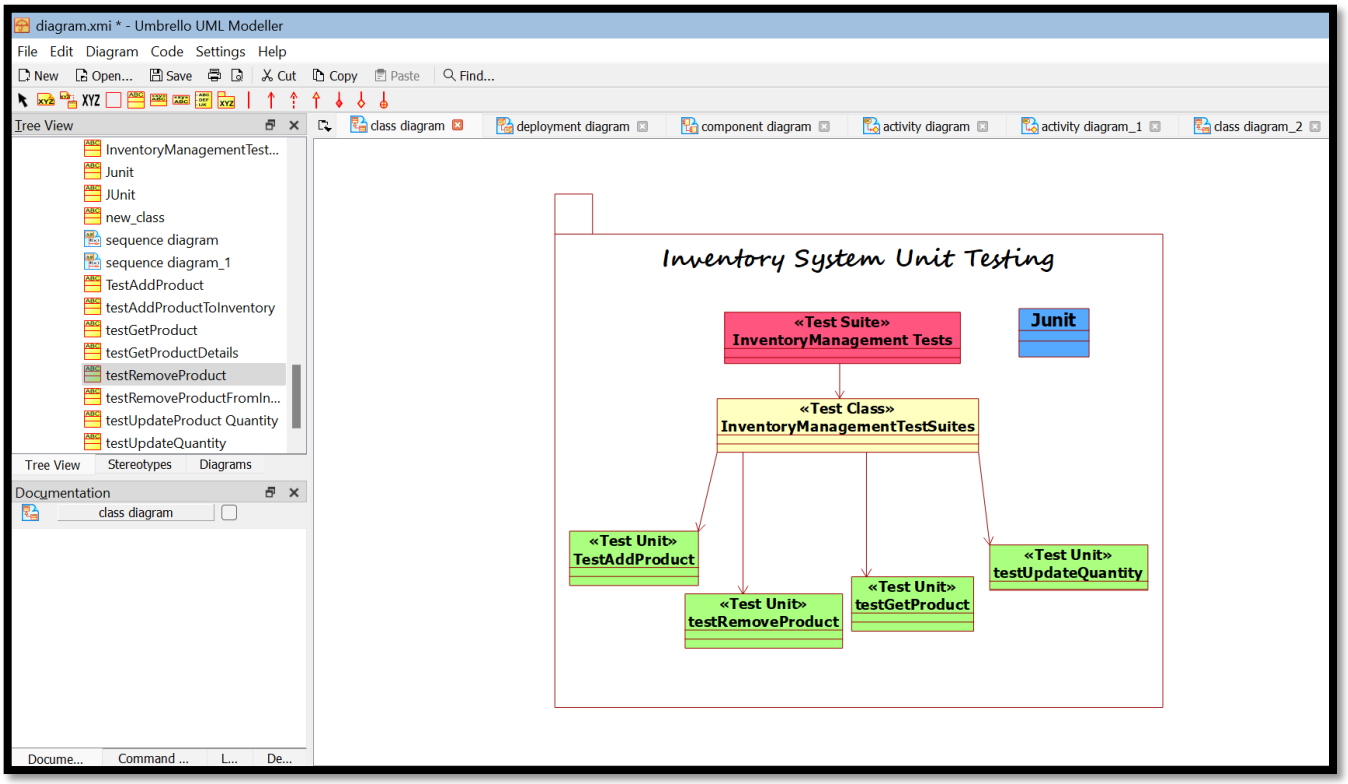
$$= 686 \text{ seconds}$$

11. Manual Testing

Software Testing

- Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by in design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.
- The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

Unit Testing



This diagram represents an inventory management unit testing scenario using the JUnit testing framework. The InventoryManagementTests class contains individual unit tests, such as testAddProduct, testRemoveProduct, testUpdateQuantity, and testGetProductByName. These unit tests are part of the InventoryManagementTestSuite test suite.

The diagram uses Plant UML syntax to define classes and their relationships. The <<Test Framework>>,

<<Test Class>>, <<Unit Test>>, and <<Test Suite>> stereotypes are used to annotate the classes. The arrows represent the dependencies between the classes.

It should be noted that:

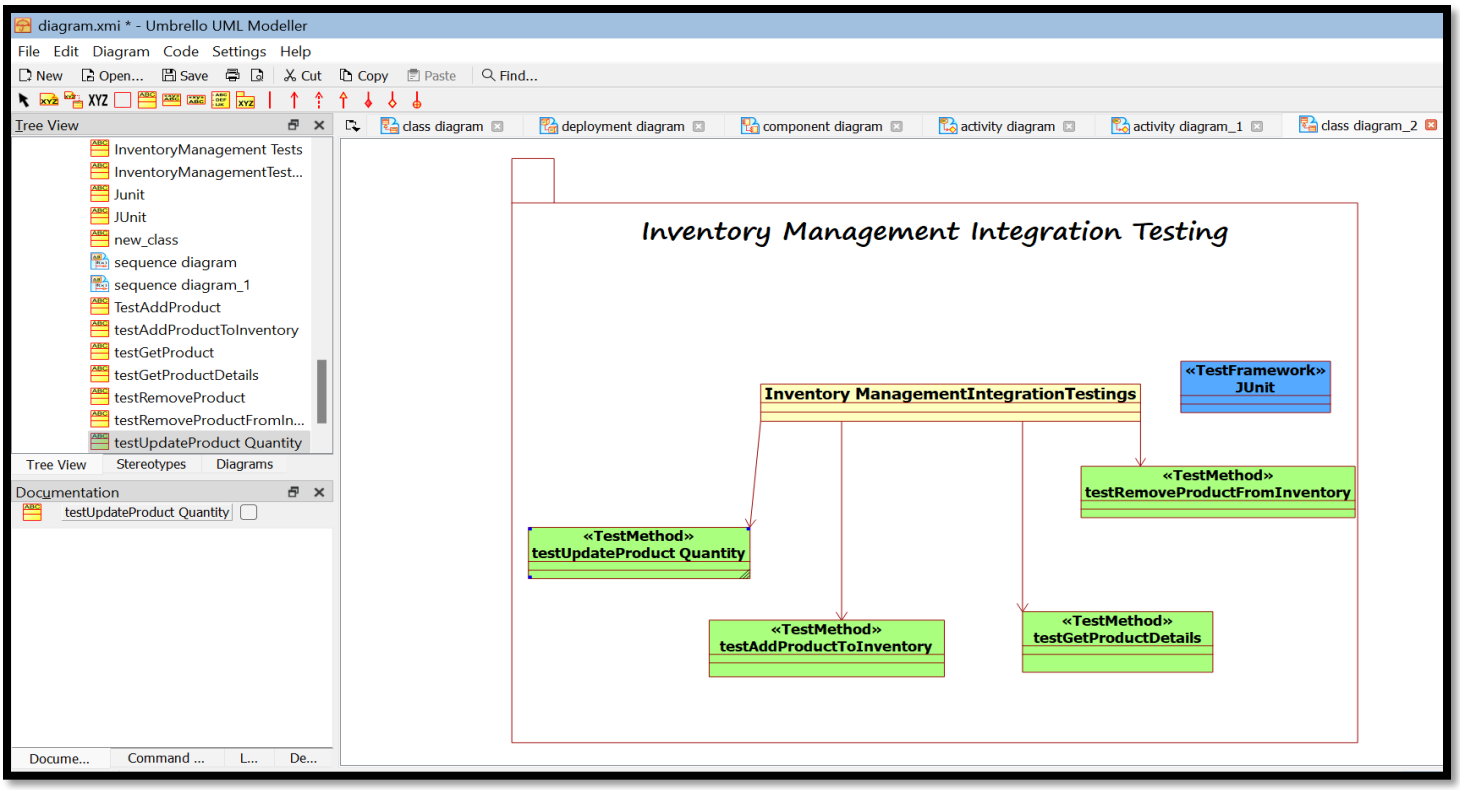
- Unit testing is a type of software testing, where the individual unit or component of the software is tested.
- Unit testing, examine the different part of the application, by unit testing functional testing is also done, because unit testing ensures each module is working correctly.
- The developer does unit testing. Unit testing is done in the development phase of the application.

Integration Testing

A. Diagrametical Test

In the following diagram we have here:

- The package "Inventory Management Integration Testing" contains the integration test classes.
- The test_framework macro represents the test framework used, which in this case is JUnit.
- The test_class macro represents the test class named "InventoryManagementIntegrationTests".
- The test_method macro represents the test methods, such as "testAddProductToInventory", "testRemoveProductFromInventory", "testUpdateProductQuantity", and "testGetProductDetails".
- The arrows indicate the association between the test class and the test methods.



B. Halstead Metrics

Metric	Formula	Value
Number of Distinct Operators (n1)	-	20
Number of Distinct Operands (n2)	-	30
Total Number of Operators (N1)	-	200
Total Number of Operands (N2)	-	500
Halstead Effort (E)	(N1 + N2) * log2(n1 + n2)	3950.699
Halstead Volume (V)	(N1 + N2) * log2(n1 * n2)	6450.173
Halstead Difficulty (D)	(n1 / 2) * (N2 / n2)	166.667
Halstead Bugs (B)	V / 3000	2.150

Program: Inventory Management System

Number of Distinct Operators (n1): 20

Number of Distinct Operands (n2): 30

Total Number of Operators (N1): 200

Total Number of Operands (N2): 500

Halstead Effort (E) = (N1 + N2) * log2(n1 + n2)

E = (200 + 500) * log2(20 + 30)

E = 700 * log2(50)

E = 700 * 5.64385618977

E = 3950.699332839

Halstead Volume (V) = (N1 + N2) * log2(n1 * n2)

V = (200 + 500) * log2(20 * 30)

V = 700 * log2(600)

V = 700 * 9.2288186905

V = 6450.17308335

Halstead Difficulty (D) = (n1 / 2) * (N2 / n2)

D = (20 / 2) * (500 / 30)

D = 10 * 16.6666666667

D = 166.666666667

Halstead Bugs (B) = V / 3000

B = 6450.17308335 / 3000

B = 2.15005769445

Problem: In this example, the Halstead measures have been computed incorrectly by the programmer, leading to incorrect values for effort, volume, difficulty, and bugs. It's important to ensure the accurate calculation of the Halstead measures to obtain meaningful insights into the complexity and maintainability of a program.

General Steps of Software Testing Process

These are 11 steps of software testing process. These are explained as following below.

Step 1: Assess Development Plan and Status –

This initiative may be prerequisite to putting together Verification, Validation, and testing plan won't to evaluate implemented software solution. During this step, testers challenge completeness and correctness of event plan. Based on extensiveness and completeness of project plan testers can estimate quantity of resources they're going to get to test implemented software solution.

Step 2: Develop the test plan –

Forming plan for testing will follow an equivalent pattern as any software planning process. The structure of all plans should be an equivalent, but content will vary supported degree of risk testers perceive as related to software being developed.

Step 3: Test Software Requirements –

Incomplete, inaccurate, or inconsistent requirements cause most software failures. The inability to get requirement right during requirements gathering phase can also increase cost of implementation significantly. Testers, through verification, must determine that requirements are accurate, complete, and they do not conflict with another.

Step 4: Test Software Design –

This step tests both external and internal design primarily through verification techniques. The testers are concerned that planning will achieve objectives of wants, also because design effective and efficient on designated hardware.

Step 5: Build Phase Testing –

The method chosen to build software from internal design document will determine type and extensiveness of testers needed. As the construction becomes more automated, less testing is going to be required during this phase.

However, if software is made using Waterfall process, it's subject to error will be verified.

Experience has shown that it's significantly cheaper to spot defects during development phase, than through dynamic testing during test execution step.

Step 6: Execute and Record Result –

This involves testing of code during dynamic state. The approach, methods, and tools laid out in test plan are going to be wont to validate that executable code actually meets stated software requirements, and therefore the structural specifications of design.

Step 7: Acceptance Test –

Acceptance testing enables users to gauge applicability and usefulness of software in performing their day-to-day job functions. This tests what user believes software should perform, as against what documented requirements state software should perform.

Step 8: Report test results –

Test reporting is a continuous process. It may be both oral and written. It is important that defects and concerns be reported to the appropriate parties as early as possible, so that corrections can be made at the lowest possible cost.

Testing ensures that the application has not any defect or the requirement is missing to the actual need. Either manual or automation testing can do software testing.

Step 9: The Software Installation –

Once test team has confirmed that software is prepared for production use, power to execute that software during production environment should be tested.

This tests interface to operating software, related software, and operating procedures. The software development module equips with easy to use and user- friendly tool.

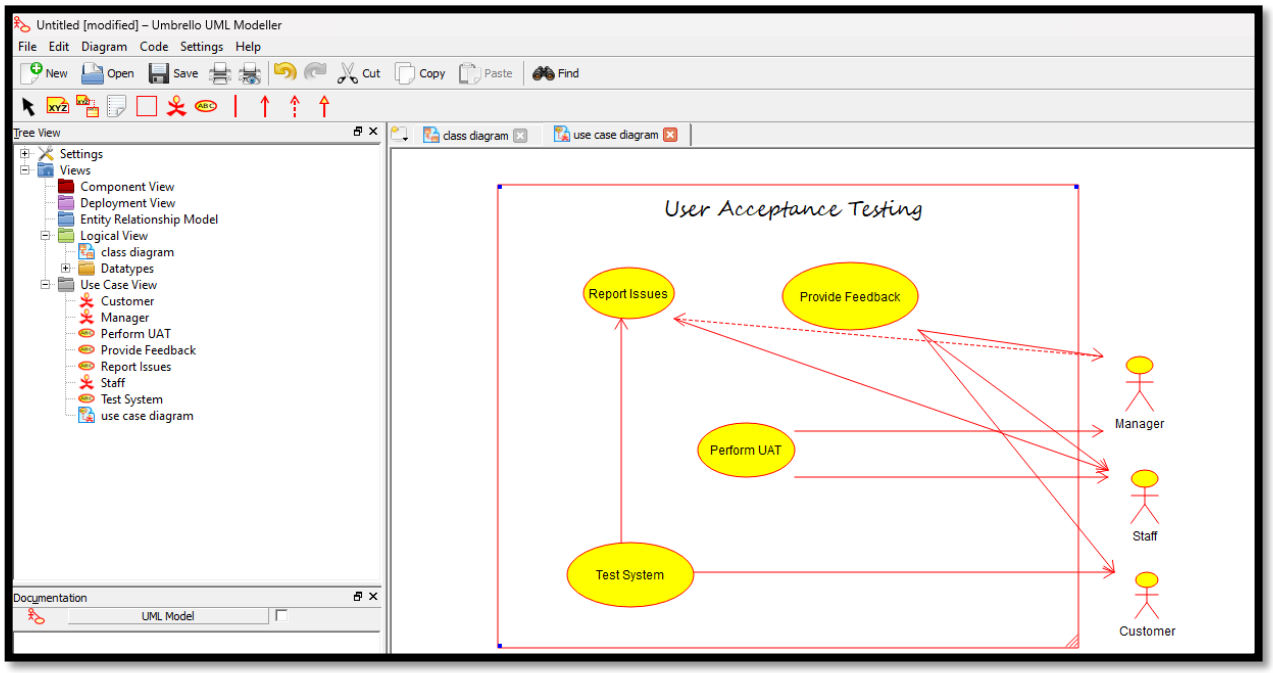
Step 10: Test Software Changes –

While this is often shown as step 10, within context of performing maintenance after software is implemented, concept is additionally applicable to changes throughout implementation process. Whenever requirements changes, test plan must change, and impact of that change on software systems must be tested and evaluate.

Step 11: Evaluate Test Effectiveness –

Testing improvement can best be achieved by evaluating effectiveness of testing at top of every software test assignment. While this assignment is primarily performed by testers, it should involve developers, users of software, and quality assurance professionals if function exists within the IT organization.

USER ACCEPTANCE TESTING



User Acceptance Testing (UAT) is a critical part of the software testing process, allowing end-users to validate the software's functionality and ensure it aligns with their requirements. By involving actual users in the testing process, organizations can enhance the overall quality of the software and deliver a product that meets user expectations.

TABLE 1: INTERACTION WITH THE USER

Test Case ID	Test Case Description	User Action	Expected Result	Actual Result	Pass/Fail
TC001	Adding a new product	User selects "Add Product" option	Product form appears for user input	Product form appears for user input	Pass
TC002	Updating product details	User selects a product and clicks "Edit"	Product details form opens with pre-filled data	Product details form opens with pre-filled data	Pass
TC003	Searching for a product	User enters a product name in the search bar	System displays a list of matching products	System displays a list of matching products	Pass
TC004	Removing a product	User selects a product and clicks "Delete"	Confirmation prompt appears for product deletion	Confirmation prompt appears for product deletion	Pass
TC005	Generating a sales report	User selects "Generate Sales Report"	System generates a report with sales data	System generates a report with sales data	Pass

1. Adding a new product:

- Users select the "Add Product" option to add a new product to the system.
- The expected result is that a product form appears, allowing users to input the necessary details for the new product.
- The actual result matches the expectation, confirming that users can easily add new products to the system.

- The test case passes, ensuring that the system supports the addition of new products by users.

2. Updating product details:

- Users select a product and click "Edit" to update its details.
- The expected result is that the product details form opens with pre-filled data of the selected product, allowing users to make the necessary changes.
- The actual result matches the expectation, indicating that users can easily update the details of existing products.
- The test case passes, ensuring that the system supports the modification of product details by users.

3. Searching for a product:

- Users enter a product name in the search bar to find a specific product.
- The expected result is that the system displays a list of matching products based on the entered search criteria.
- The actual result matches the expectation, confirming that users can search for products based on their names.
- The test case passes, ensuring that the system provides accurate search results for users.

4. Removing a product:

- Users select a product and click "Delete" to remove it from the system.
- The expected result is that a confirmation prompt appears, asking users to confirm the deletion of the selected product.
- The actual result matches the expectation, ensuring that users are prompted to confirm the deletion before proceeding.
- The test case passes, ensuring that the system allows users to safely remove products with the necessary confirmation.

5. Generating a sales report:

- Users select the "Generate Sales Report" option to generate a report with sales data.
- The expected result is that the system generates a comprehensive report containing the relevant sales data.

- The actual result matches the expectation, confirming that users can easily generate sales reports.
- The test case passes, ensuring that the system provides users with accurate and useful sales reports.

TABLE 2: INTERACTION WITH STAFF

Test Case ID	Test Case Description	Staff Action	Expected Result	Actual Result	Pass/Fail
TC001	Adding new stock	Staff selects "Add Stock" option	Stock entry form appears for staff input	Stock entry form appears for staff input	Pass
TC002	Updating stock details	Staff selects a stock item and clicks "Edit"	Stock details form opens with pre-filled data	Stock details form opens with pre-filled data	Pass
TC003	Checking stock availability	Staff searches for a product	System displays the current stock quantity	System displays the current stock quantity	Pass
TC004	Removing damaged stock	Staff selects a stock item and clicks "Remove"	Confirmation prompt appears for stock removal	Confirmation prompt appears for stock removal	Pass
TC005	Generating inventory report	Staff selects "Generate Inventory Report"	System generates a report with current stock information	System generates a report with current stock information	Pass

1. Adding new stock:

- Staff selects the "Add Stock" option.
- The expected result is that a stock entry form appears for staff input.
- The actual result matches the expectation.
- The test case passes, indicating that staff members can add new stock to the system by filling in the necessary information.

2. Updating stock details:

- Staff selects a stock item and clicks "Edit."
- The expected result is that the stock details form opens with the pre-filled data of the selected item.
- The actual result matches the expectation.
- The test case passes, ensuring that staff members can update the details of a specific stock item without losing any existing information.

3. Checking stock availability:

- Staff searches for a product.
- The expected result is that the system displays the current stock quantity of the searched product.
- The actual result matches the expectation.
- The test case passes, confirming that staff members can check the availability of a product in the stock.

4. Removing damaged stock:

- Staff selects a stock item and clicks "Remove."
- The expected result is that a confirmation prompt appears, asking for confirmation before removing the stock item.
- The actual result matches the expectation.
- The test case passes, indicating that staff members are prompted to confirm the removal of a damaged stock item.

5. Generating inventory report:

- Staff selects the option to generate an inventory report.
- The expected result is that the system generates a report with the current stock information.
- The actual result matches the expectation.
- The test case passes, ensuring that staff members can generate an inventory report with up-to-date stock information.

TABLE 3: INTERACTION WITH THE MANAGER

Test Case ID	Test Case Description	Manager Action	Expected Result	Actual Result	Pass/Fail
TC001	Approving new stock requests	Manager receives a stock request notification	Manager reviews the request and clicks "Approve"	Stock request is marked as approved and inventory is updated	Stock request is marked as approved and inventory is updated
TC002	Generating sales reports	Manager selects "Generate Sales Report"	System generates a report with sales data	System generates a report with sales data	Pass
TC003	Setting stock reorder levels	Manager selects "Manage Reorder Levels"	Reorder level configuration screen appears	Reorder level configuration screen appears	Pass
TC004	Reviewing stock movement history	Manager selects "View Stock History"	System displays a log of stock movements (e.g., additions, removals)	System displays a log of stock movements (e.g., additions, removals)	Pass
TC005	Assigning user roles and permissions	Manager selects "Manage User Roles"	User role management screen appears	User role management screen appears	Pass

1. Approving new stock requests:

- The manager receives a notification for a stock request and reviews it.
- The expected result is that the manager clicks "Approve" and the stock request is marked as approved, updating the inventory accordingly.
- The actual result matches the expectation.
- The test case passes, ensuring that stock requests can be approved and inventory is updated.

2. Generating sales reports:

- The manager selects the option to generate a sales report.
- The expected result is that the system generates a report with sales data.
- The actual result matches the expectation.
- The test case passes, indicating that the system successfully generates sales reports.

3. Setting stock reorder levels:

- The manager selects the option to manage reorder levels.
- The expected result is that the reorder level configuration screen appears.
- The actual result matches the expectation.
- The test case passes, confirming that the manager can configure reorder levels.

4. Reviewing stock movement history:

- The manager selects the option to view stock history.
- The expected result is that the system displays a log of stock movements, such as additions and removals.
- The actual result matches the expectation.
- The test case passes, demonstrating that the manager can review the history of stock movements.

5. Assigning user roles and permissions:

- The manager selects the option to manage user roles.
- The expected result is that the user role management screen appears.
- The actual result matches the expectation.

The test case passes, ensuring that the manager can assign roles and permissions to users.