REPUBLIC OF CAMEROON

Peace- Work- Fatherland

African Institute of Computer Science

(AICs)

REPUBLIQUE DU CAMEROUN

Paix- Travail- Patrie

Institute Africain D'informatique

(IAI)

CLASS: BA2B

Object Oriented Programming: JAVA

THEME: Grocery Store Stock Management App

GROUP 35 MEMBERS

- **TEDJIOZEM CHRIST 30%**
- **NGONGANG TEKEDO LANDRY 70%**

Course Supervisor:

Mr. Fomekong

Academic Year 2024-2025

TABLE OF CONTENT

I - INTRODUCTION

The Grocery Store Stock Management App is a software application designed to streamline the management of stock in a grocery store. The app allows store managers and administrators to efficiently track, update, and manage inventory, ensuring that the store operates smoothly and avoids stockouts or overstocking. This report provides an overview of the app's features, analysis and implementation.

II - OBJECTIVES

The primary objectives of the Grocery Store Stock Management System are:

- To enable stock managers to add, update, delete, and view stock.
- To provide an authentication system with different user roles (admin and stock manager).
- To facilitate smooth database integration with MySQL.
- To ensure data accuracy and security in stock management operations.

III – ANALYSIS

1. Use Case Diagram

a) Definition

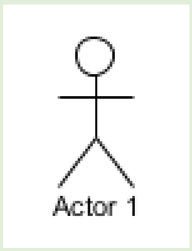
A use case diagram describes the high-level functions and scope of a system. A use case is a specification of behaviour. The main objectives of the use case diagram are:

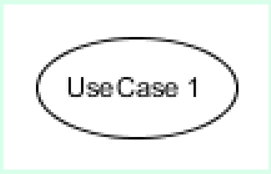
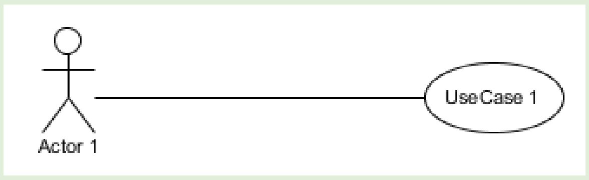
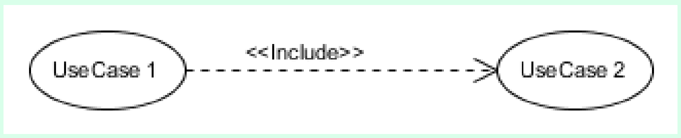
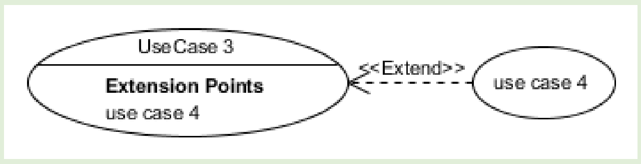
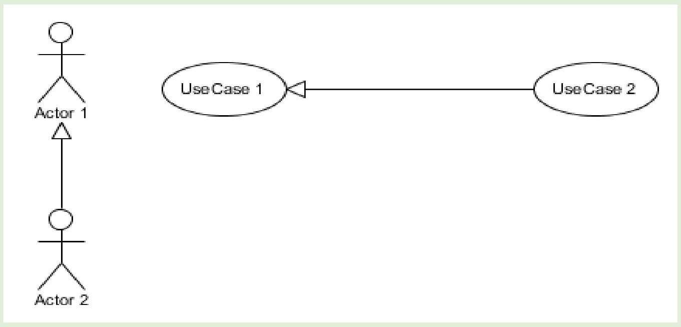
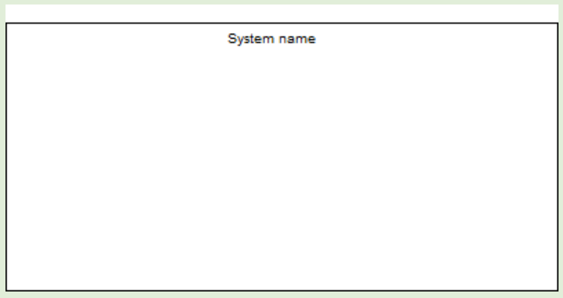
• Provide a high-level view of the system

• Identify the functions of the system.

Use case diagrams are complemented by textual descriptions of each use case, providing greater detail and clarity about their functionalities and interactions.

b) Use case Diagram Components

Element	Description and main	Notation
Actor	Represents an entity that directly interacts with the system. The actor is what performs the different possible actions of the system.	

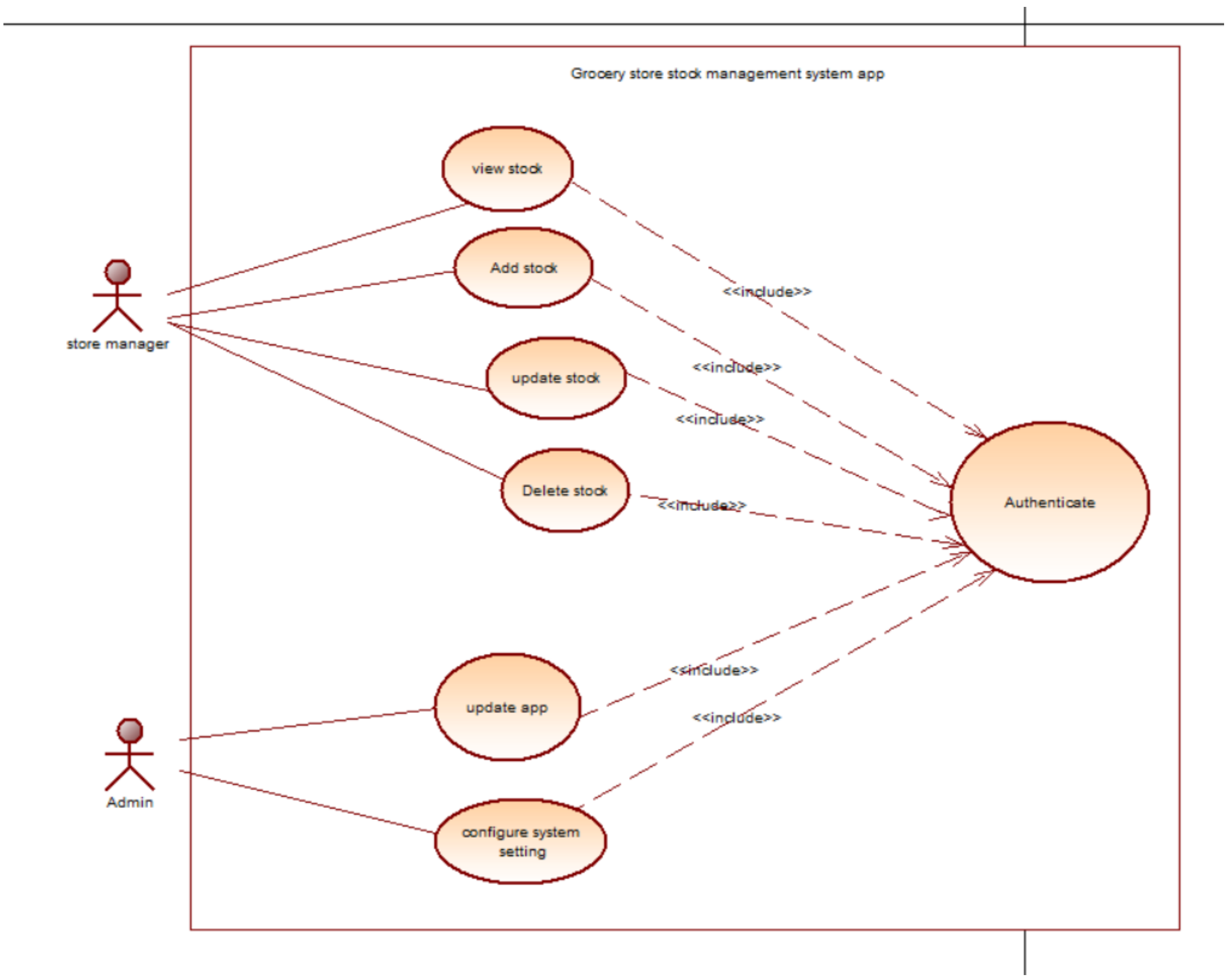
Use case	A use case represents a functionality of the system. It is an action that can be performed by an actor.	
Association	It indicates that an actor take part in a use case	
Include	An inclusion denotes that an included action must be performed before the including action can be performed.	
Extend	An extension denotes that an extending action may be performed while an extended action is being performed.	 Use Case 3 Extension Points use case 4 <<Extend>> use case 4
Generalization	This shows that a uses case is a kind of another. This relation also permits to decompose a complex case into smaller and simple cases.	
System	It is a container of use cases which interact with the external actors.	

c) The Actor of our system

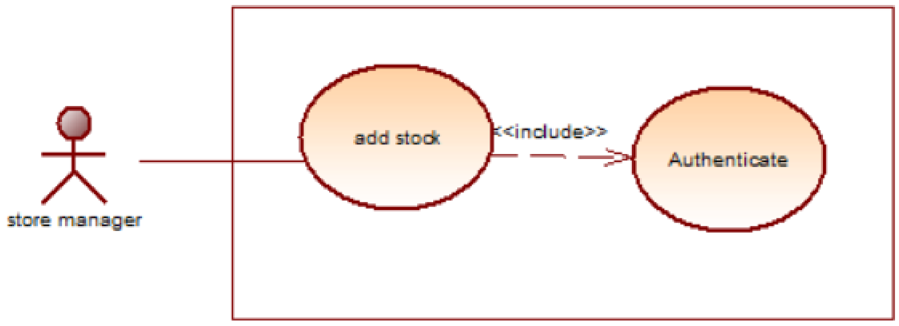
After the study of the current system, we identified the actors listed in the table below.

Store Manager	Responsible of viewing and adding stock, updating and deleting.
Administrator	Responsible of updating the application and configuring system settings.

d) General Use case

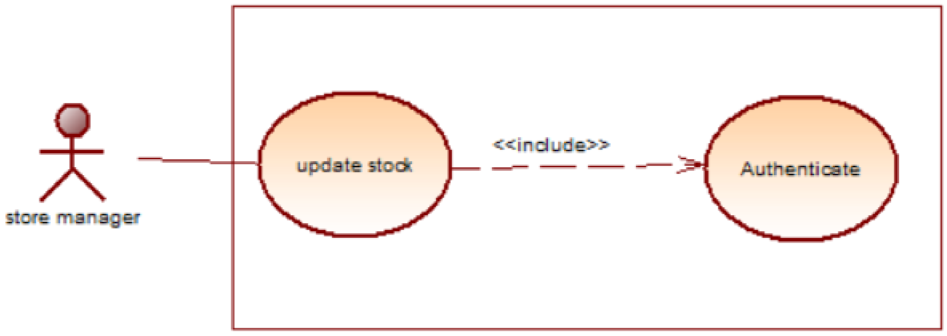


e) Specific Use case of add stock



Specific use case for add stock

f) Specific Use case Update stock



specific use case for update stock

g) Textual description of add stock

Title	Add stock
Actors	Store manager
Trigger	The user (store manager clicks on the add button
Summary	To permit the store manager to add stock
Pre-conditions	-The database connection must be active - The manager must enter product details
Nominal scenario	1.The store manager enters product details and clicks the create button. 2.The system validates the input fields. 3. If valid, the system inserts the new product into the database. 4.The products appears in the stock table. 5.The system confirms the successful addition.
Alternative scenario	If input is invalid the system shows an error message.
Post-condition of success	The product is added to stock
Post-condition of failure	The product is not added to stock

h) Textual description of update stock

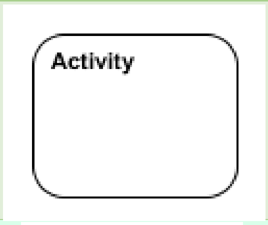
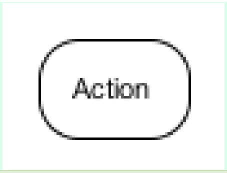
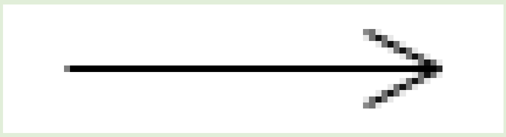
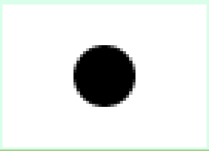
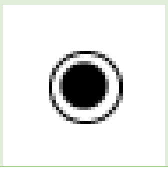
Title	update stock
Actors	Store manager
Summary	To permit the store manager to update stock
Trigger	The store manager clicks on the update
Pre-conditions	- The product must already exist in the database. -The manager must select a product from the table.
Nominal scenario	1. The store manager selects a product from the stock list, modifies the necessary fields (e.g name, quantity, price) and clicks the "update" button. 2. The system verifies input field and sends the data to the DBMS 3. The DBMS execute query and send result. 4. The system updates the corresponding entry in the database. 5. The stock table refreshes to reflect the changes.
Alternative scenario	If the input field is incorrect, the system rejects the update and shows an error message.
Post condition of success	The product updates in the table and database.
Post condition of failure	The product doesn't update.

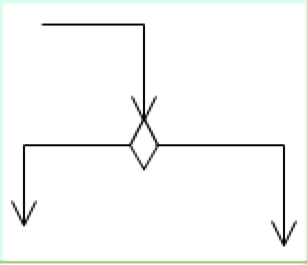
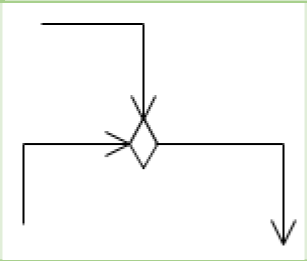
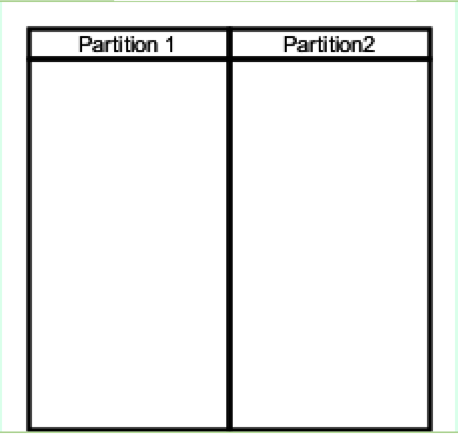
2. Activity Diagram

a) Definition

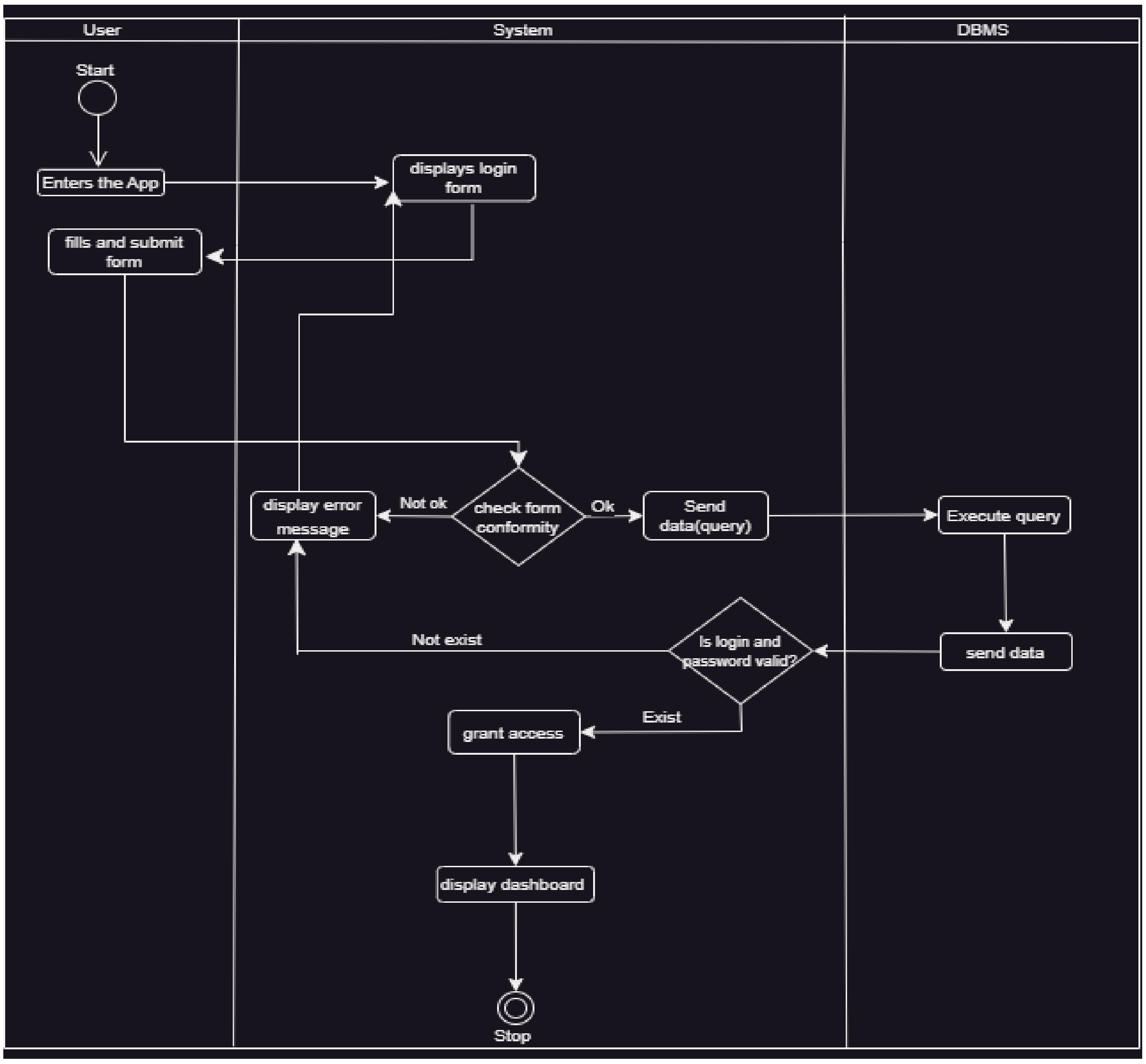
An activity diagram is a graphical representation that illustrates the sequence of activities within a system. It provides a clear description of the flow from a starting point to an endpoint, incorporating all possible decisions and actions throughout. Activity diagrams consolidate the specifications of use cases and can also highlight instances where parallel processing may occur during the execution of certain activities.

b) Activity Diagram Components

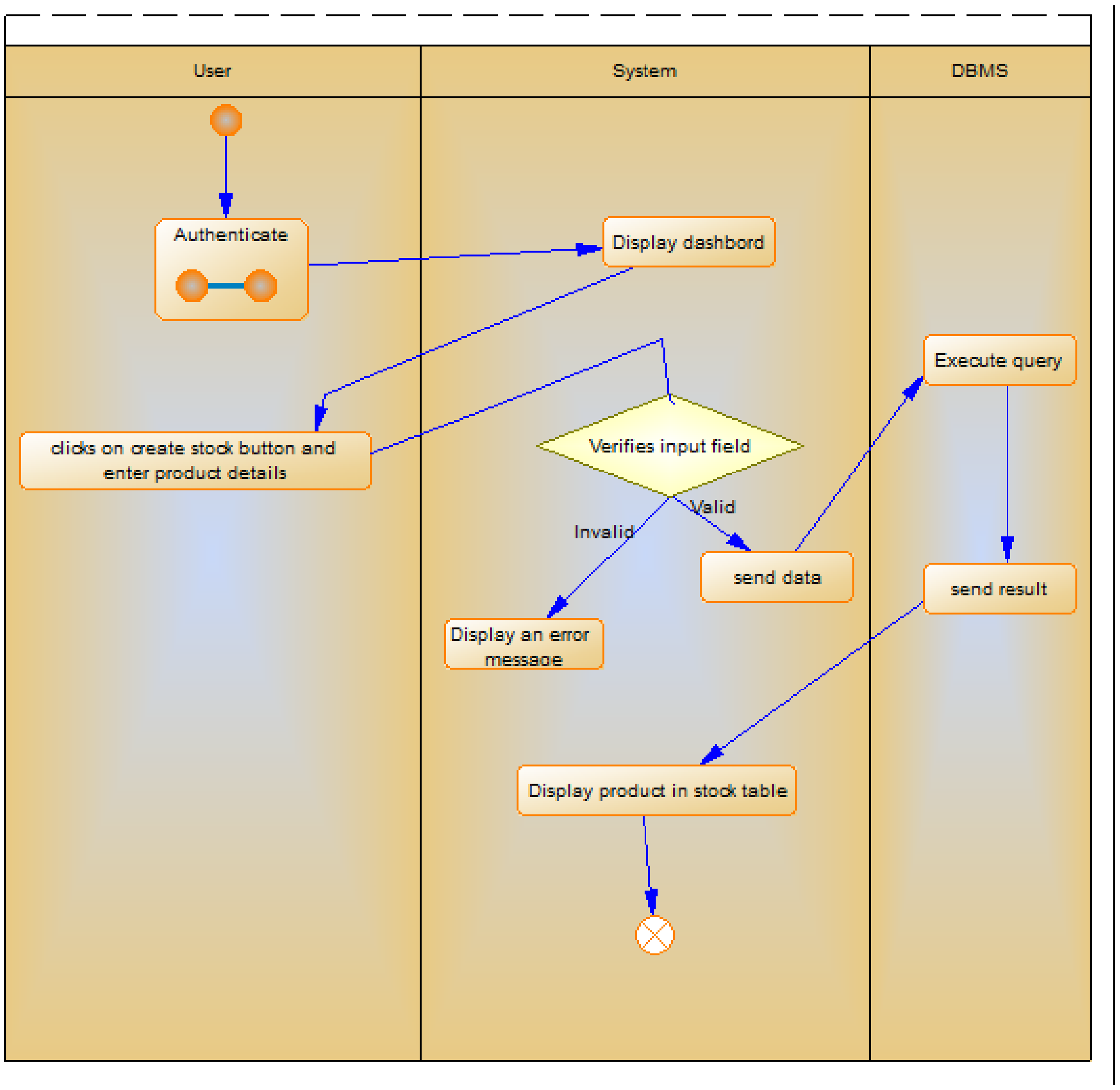
Element	Description and main properties	Notation
Activities	Use to represent a set of actions.	
Action	Represent a task to be performed.	
Control flow	A directed connection between two activity nodes through which tokens may flow.	
Initial node	Shows the beginning of an activity or set of actions.	
Final node	Stops all controls and object flows in an entire activity.	

Decision node	Represents a test condition that slits an incoming activity edge into opposite outgoing activity edges.	
Merge node	Reunite different decision paths created using a decision node.	
Partition	A way of grouping activities performed by the same actor in an activity diagram or to group actions in the same thread.	

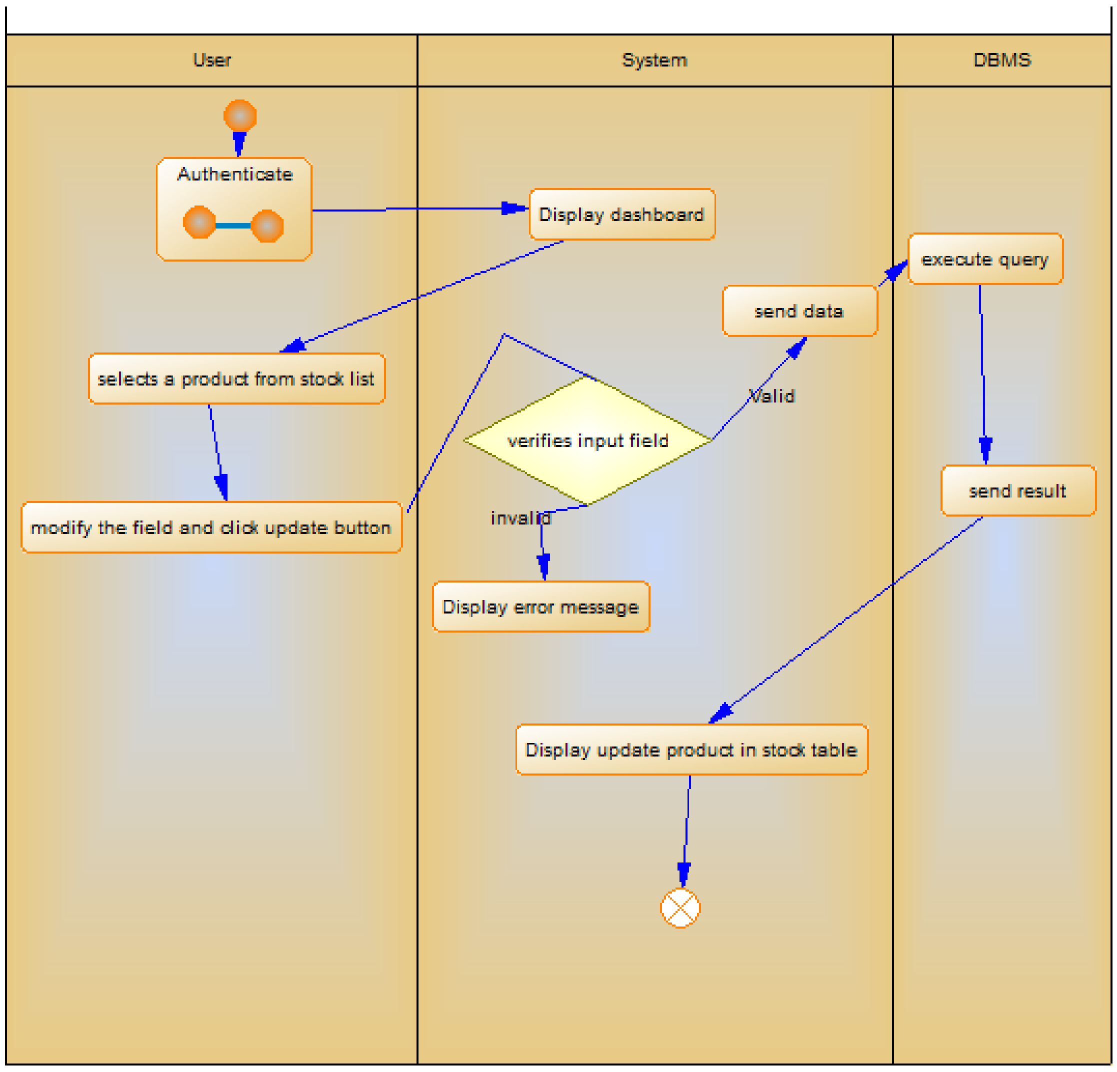
c) Authenticate Activity Diagram



d) Add stock Activity Diagram



e) Update stock Activity Diagram



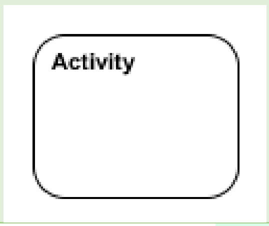
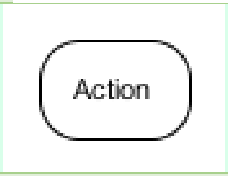
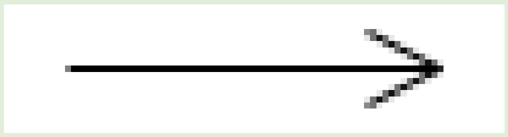
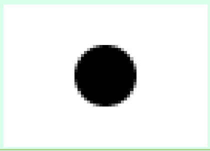
Activity diagram of update stock

3. Sequence Diagram

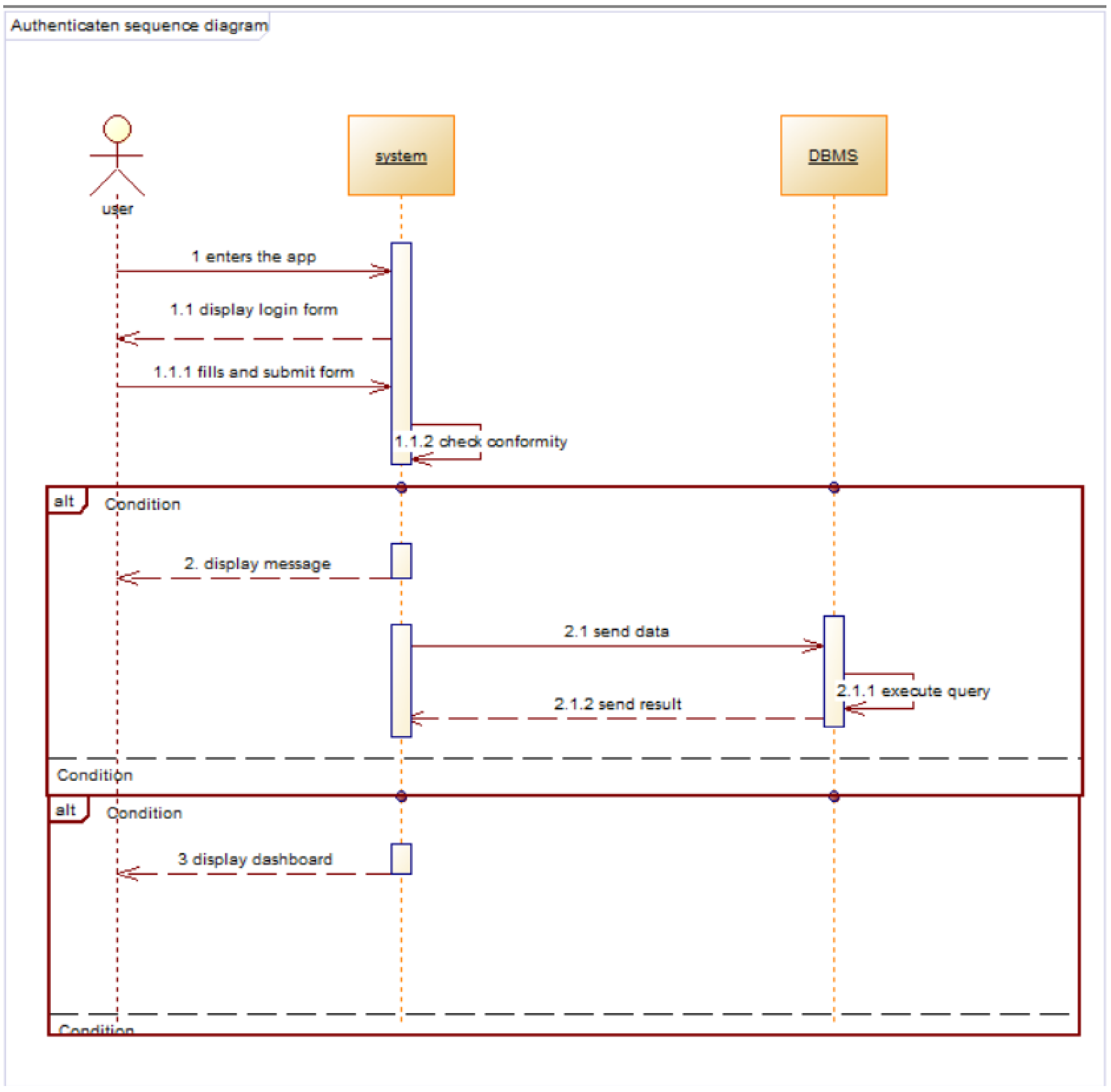
a) Definition

A sequence diagram is a form of interaction diagram, which sows objects as lifelines running down the page and with their interactions over time represented as messages drawn as arrows from the source lifeline to the target lifeline. Sequence diagrams are good at showing which object communicates with which other objects and what messages trigger those communications. Sequence diagrams are not intended for showing complex procedural logic.

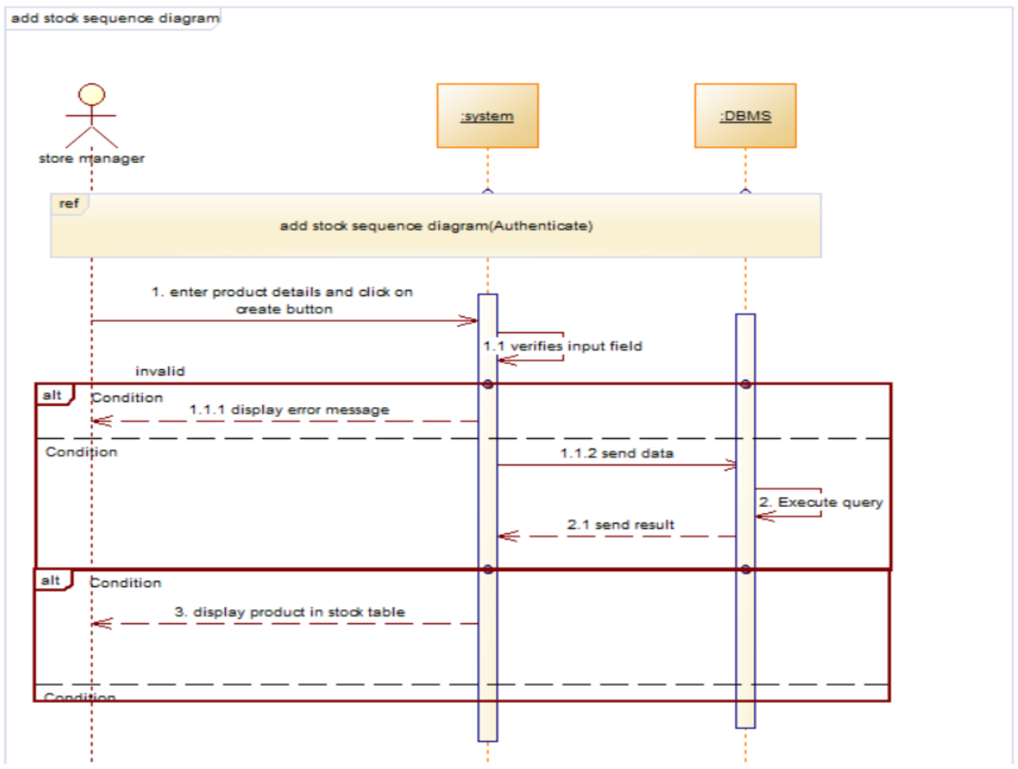
b) Sequence Diagram Components

Element	Description and main properties	Notation
Lifeline	It represents an individual participant in a sequence diagram, it is position at the top of the diagram.	
Combined fragment	It represents a choice of behaviour in which at most one operand will be chosen.	
Messages	These are arrows which represent direction of message flow. We have the synchronous, the asynchronous and the self-messages.	
Activation	It describes the time period in which an operation is performed by an element.	

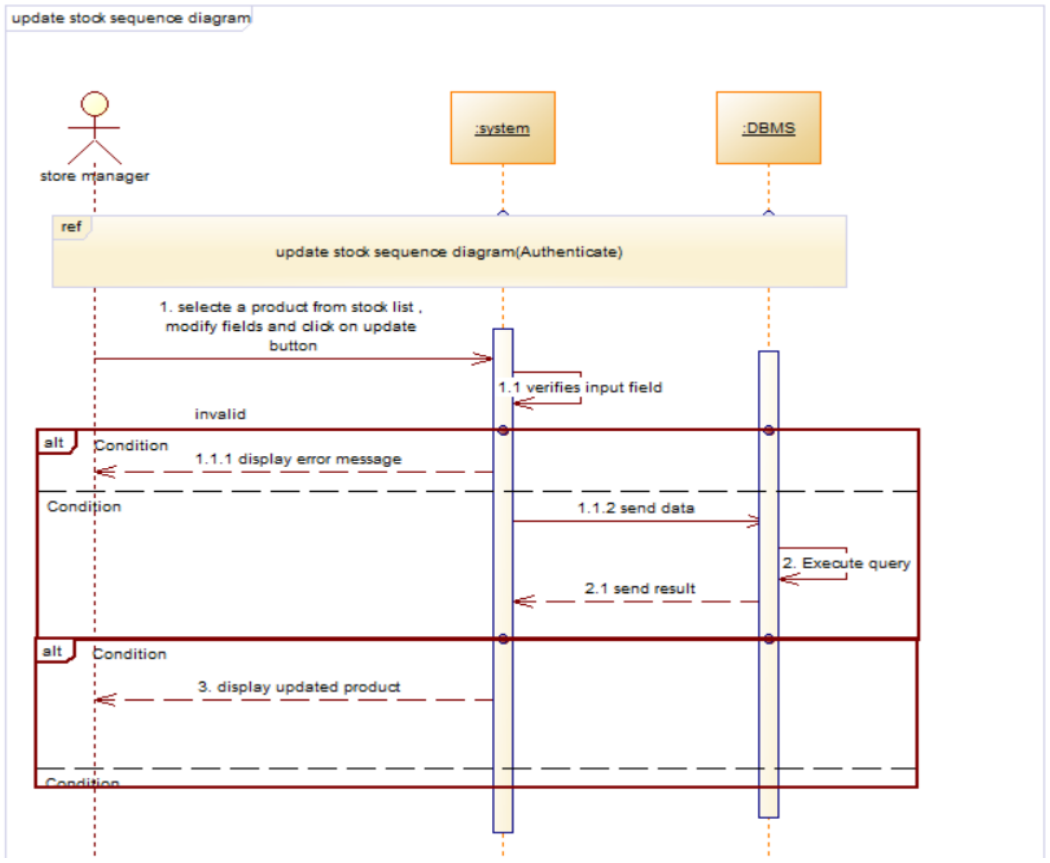
c) Authenticate Sequence Diagram



d) Add stock Sequence Diagram


add stock sequence diagram
store manager
:system
:DBMS
ref
add stock sequence diagram(Authenticate)
1. enter product details and click on create button
1.1 verifies input field
invalid
alt
Condition
1.1.1 display error message
Condition
1.1.2 send data
2. Execute query
2.1 send result
alt
Condition
3. display product in stock table
Condition

e) Update stock Sequence Diagram

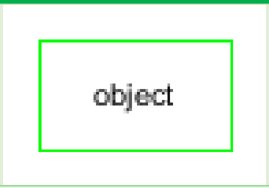
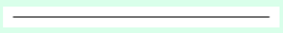
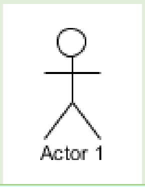
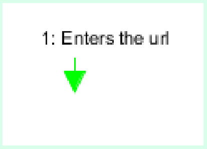

update stock sequence diagram
store manager
:system
:DBMS
ref
update stock sequence diagram(Authenticate)
1. selecte a product from stock list , modify fields and click on update button
1.1 verifies input field
invalid
alt
Condition
1.1.1 display error message
Condition
1.1.2 send data
2. Execute query
2.1 send result
alt
Condition
3. display updated product
Condition

4. Communication Diagram

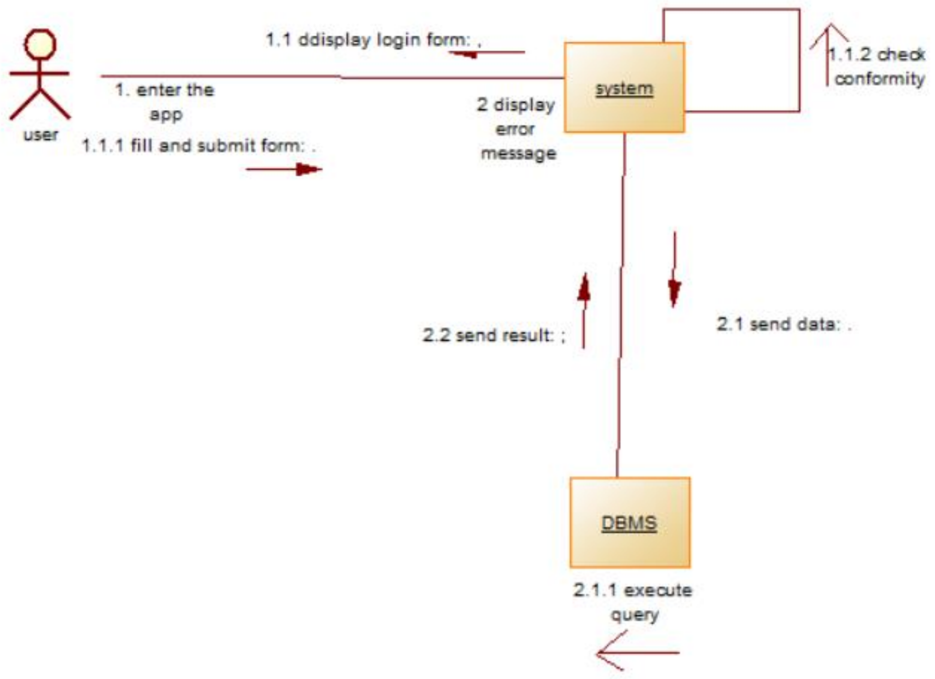
a) Definition

It is a diagram which is used to show the relationship between the actors of a system, both the sequence and the communication diagrams represent the same information but differently. Instead of showing the flow message, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.

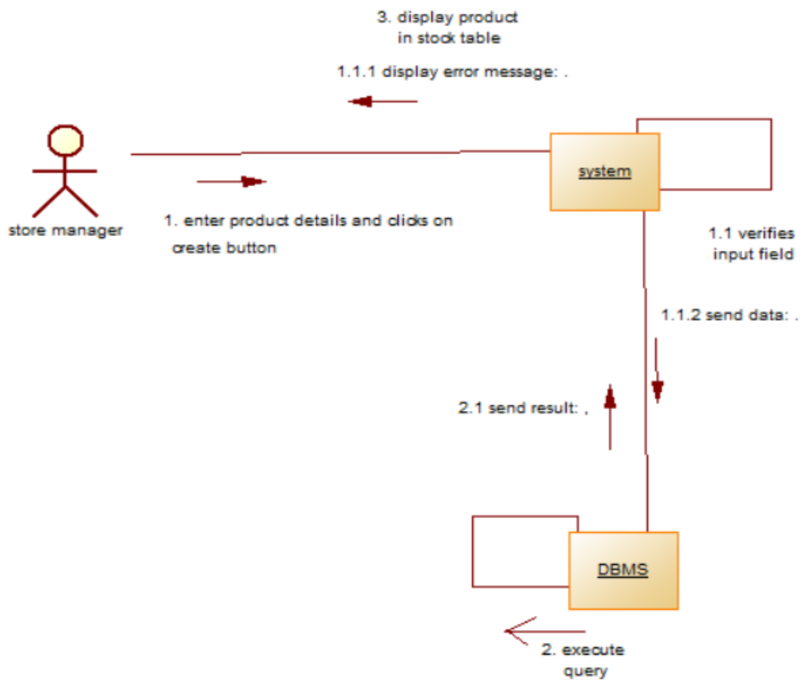
b) Communication Diagram Components

Element	Description and main properties	Notation
Object	An object represents an individual participant in the interaction Conversion**.**	
Link	It represents a choice of behaviour in which at most one operand will be chosen.	
Actor	A role plays by an entity that interacts with the subjects.	
Message	It describes the time period in which an operation is performed by an element.	

c) Authenticate Communication Diagram

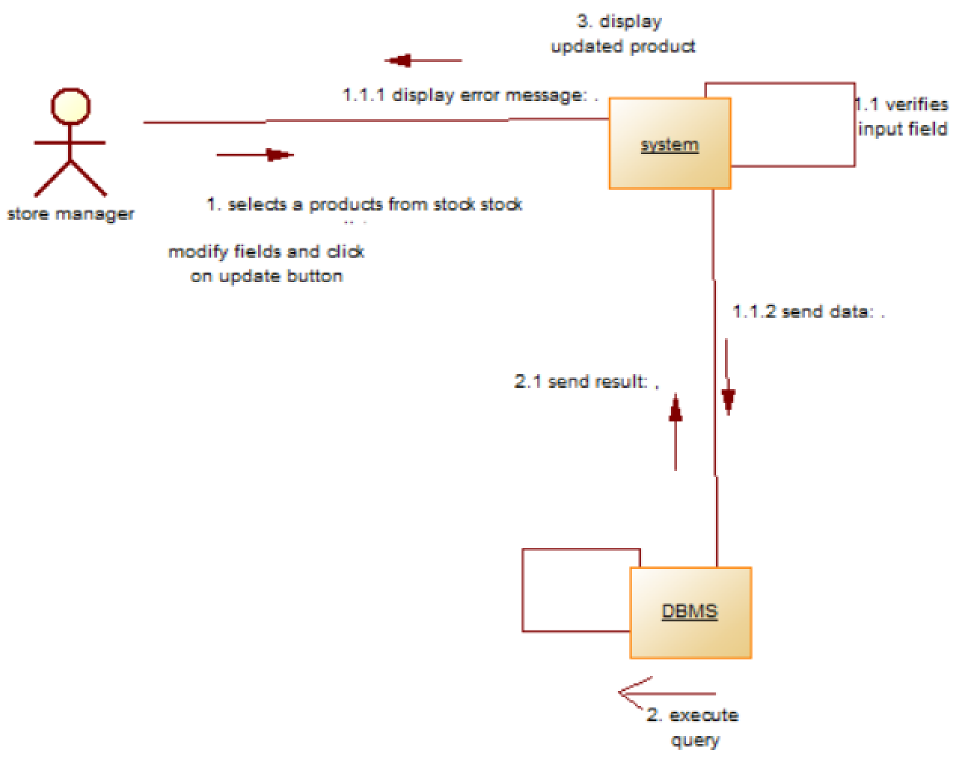


d) Add stock Communication Diagram



3. display product in stock table
1.1.1 display error message: .
store manager
1. enter product details and clicks on create button
system
1.1 verifies input field
1.1.2 send data: .
2.1 send result: ,
DBMS
2. execute query

e) Update stock Communication Diagram

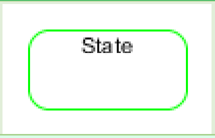
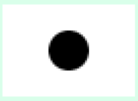
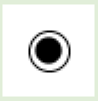
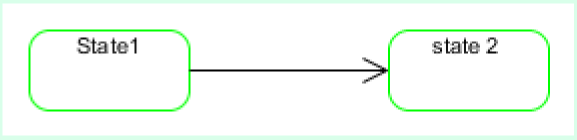
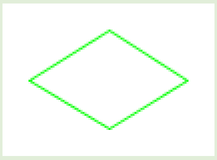
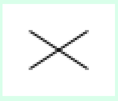
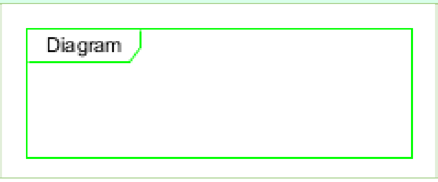

3. display updated product
1.1.1 display error message: .
1.1 verifies input field
system
store manager
1. selects a products from stock stock
modify fields and click on update button
1.1.2 send data: .
2.1 send result: ,
DBMS
2. execute query

5. State machine Diagram

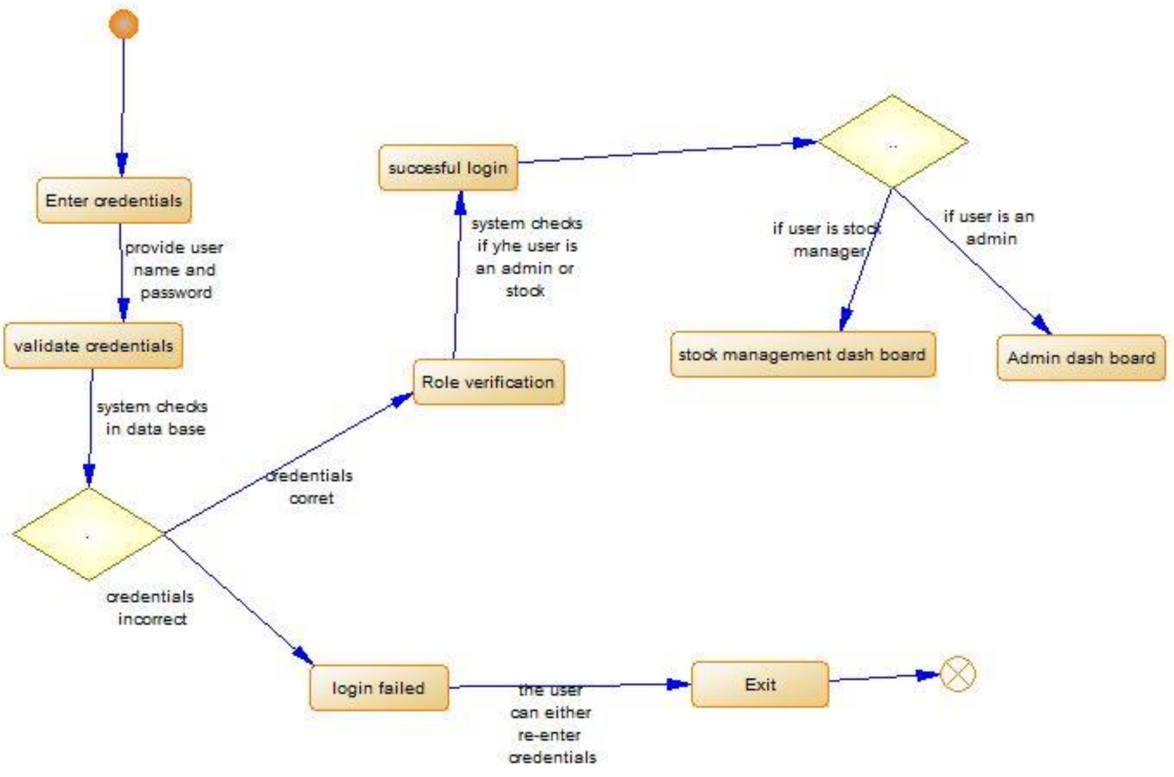
a) Definition

A state machine diagram, also known as a state diagram, illustrates the behaviour of an individual object in response to a sequence of events within a system. It depicts the dynamic flow of control and transitions between different states that the object can assume.

b) Components of a State machine diagram

Element	Description and main properties	Notation
State	Models a situation during which a certain invariant condition holds.	
Initial State	It represents a default vertex, that is, a source for a single transaction to the default or composite state.	
Final State	A state specifying that the enclosing region is complete.	
Transition	A direction relation between a source and a target vertex.	
Choice pseudo-State	A diamond symbol that indicates a dynamic condition with branched potential results	
Terminate	Implies that the execution of a state by means of its context is terminated.	
Diagram Overview	A placeholder for the linked states in a state machine diagram	

c) Login process State machine diagram

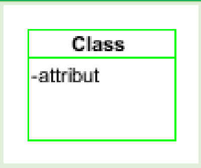
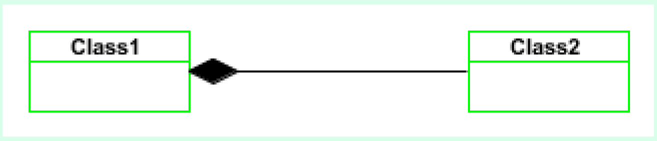
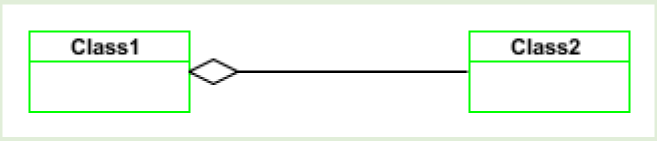
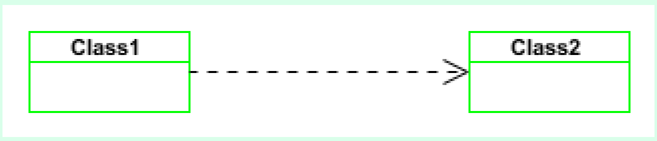
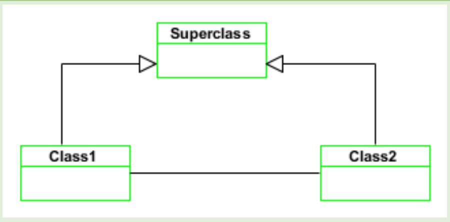
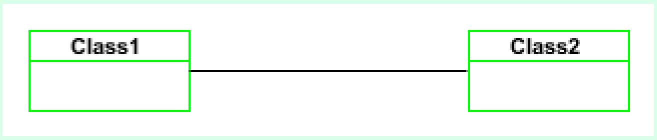


6. Class Diagram

a) Definition

A class diagram is a static diagram that provides a visual representation of the static view of an application. It is used not only for visualizing and documenting various aspects of the system but also for constructing executable code. The class diagram describes the attributes, operations and constraints of classes in the system. Its purpose is to model the static structure of an application.

b) Class Diagram Components

Element	Description and main properties	Notation
Class	A class is an element that defines the attributes and behaviours that an object is able to generate	
Composition	If a parent of a composite is deleted, usually, all of its parts are deleted with it.	
Aggregation	If the parent of the aggregate is deleted, usually the children are not deleted.	
Dependency	It existed between two classes, if one changes it may cause the change in the order, but the other way around.	
Generalization	it a relationship between a whole thing (called superclass) and a more specific thing (called subclass)	
Association	It is a general type of relationship between elements, it may include cardinality, roles etc.	

c) System Class Diagram

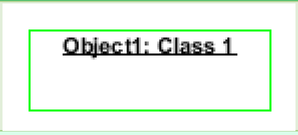
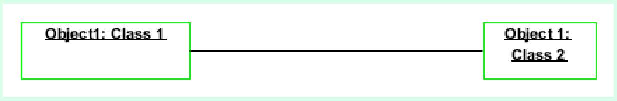
SIR, I DON'T UNDERSTAND WHY BUT WHEN I PASTE THE CLASS DIAGRIAM ON THIS PAGE IS DOESN'T SHOW COMPLETELY. REASON WHY I PASTED IT ON PAGE 25.

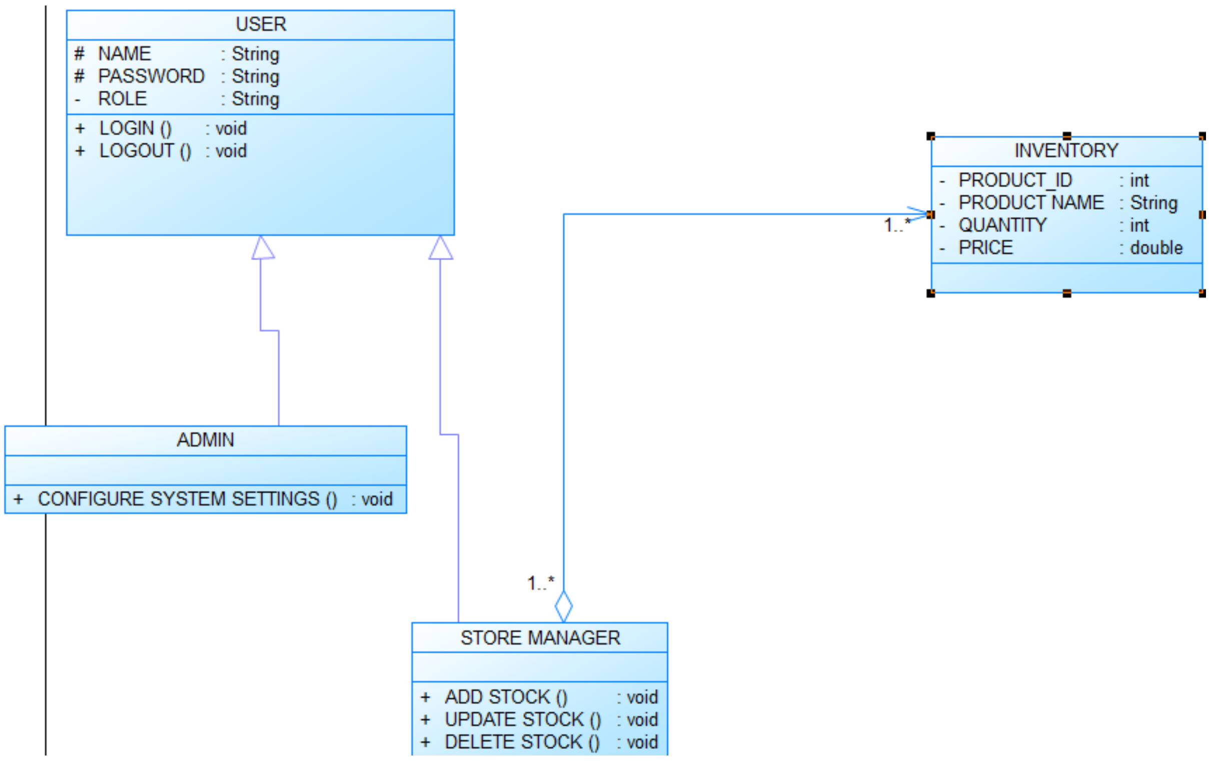
7. Object Diagram

a) Definition

An object diagram in UML is a diagram that shows a complete or partial view of the structure of a modelled system at a specific time. It emphasizes a particular set of objects and their attributes, as well as the relationships between these instances. A series of correlated object diagrams offer insight into how a system's view is expected to evolve over time.

b) Object Diagram Components

Element	**Description and main properties**	**Notation**
Object	An object is an instance of a class.	
Link	It is an instance of an association	



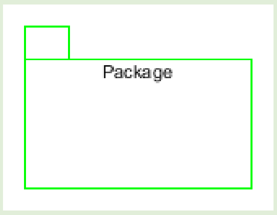
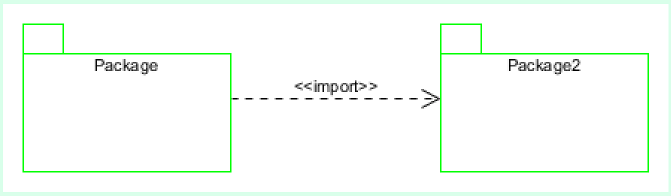
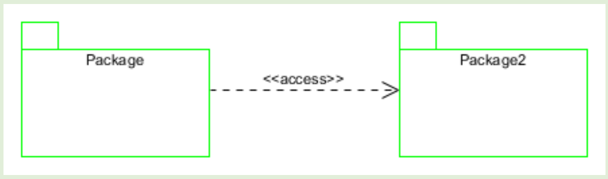
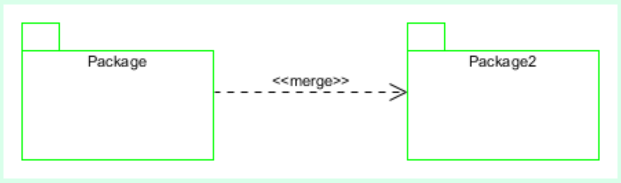
SYSTEM CLASS DIAGRAM

8. Package Diagram

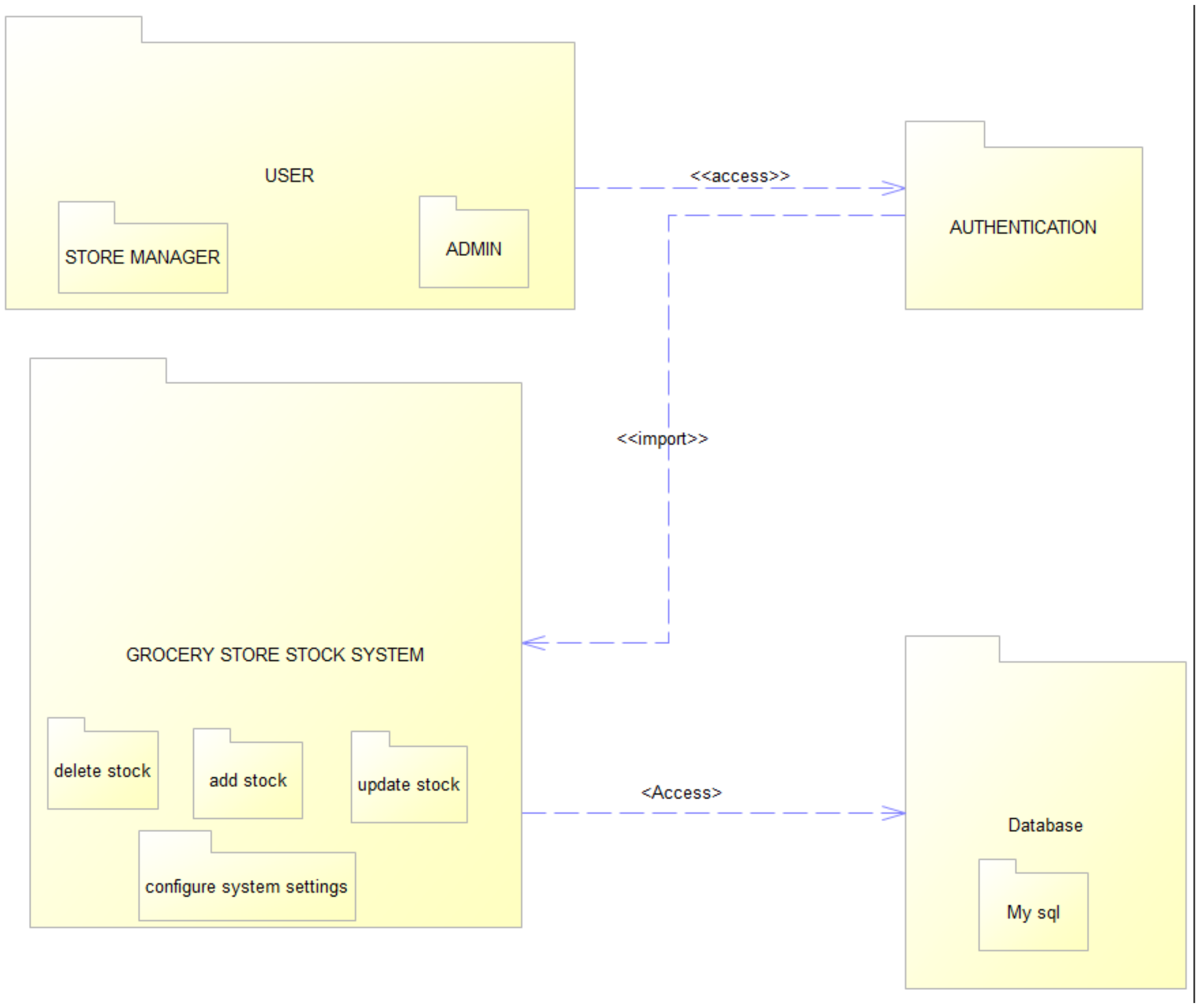
a) Definition

A package diagram is a structural diagram that depicts the organizational and arrangement of various model elements in the form of packages. It provides a visual representation of how related UML elements such as classes, diagrams, or even other packages are grouped together.

b) Package Diagram Components

Element	Description and main	Notation
Package	A package is a namespace use to group related elements; it is a mechanism used to group elements into a better structure in	
Package import	A relationship Indicate that, functionality has been imported from one package to another.	
Package access	A relationship Indicates that one package requires assistance from the function of another package	
Package merge	It is a relationship which shows that, the functionality of two packages are combines to a single function.	

c) System Package Diagram

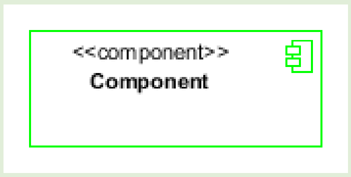
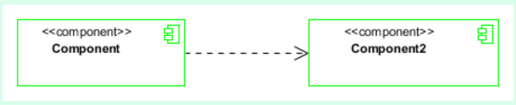
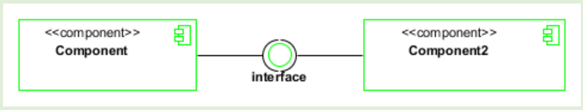
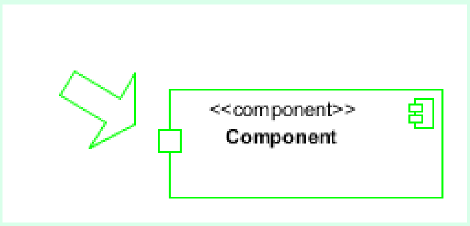


9. Component Diagram

a) Definition

Component diagrams are utilized to depict the physical elements of a system. These elements include executables, libraries, files, documents, and more, which reside within a node. It is important to note that while the component diagram does not illustrate the system's functionality, it portrays the components involved in enabling those functionalities.

b) Component Diagram Components

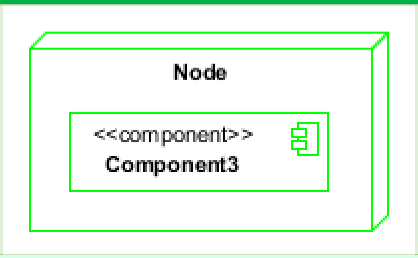
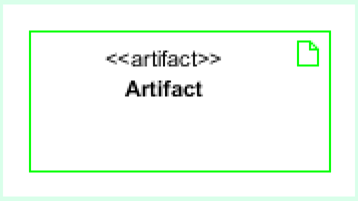
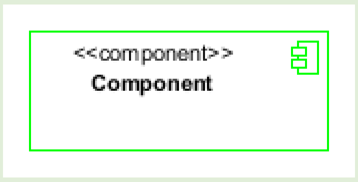
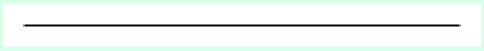
Element	Description and main properties	Notation
A component	A component is an abstract logical unit block of a system.it is represented as a rectangle with smaller rectangle in the upper right corner which saves as it icon for recognition.	
Dependency	Dependency is a directed relationship which is used to show that some components are dependent on others for their correct functioning.	
Interface	An interface is a circle or a semi-circle attached to a stick which looks like a lollipop. It describes groups of operations provided or required by components.	 <<component>> Component interface <<component>> Component2
Port	A port (represented by a small square at the end of a required or provided interface) is used when the components delegate the interfaces to an internal class.	

10.Deployment Diagram

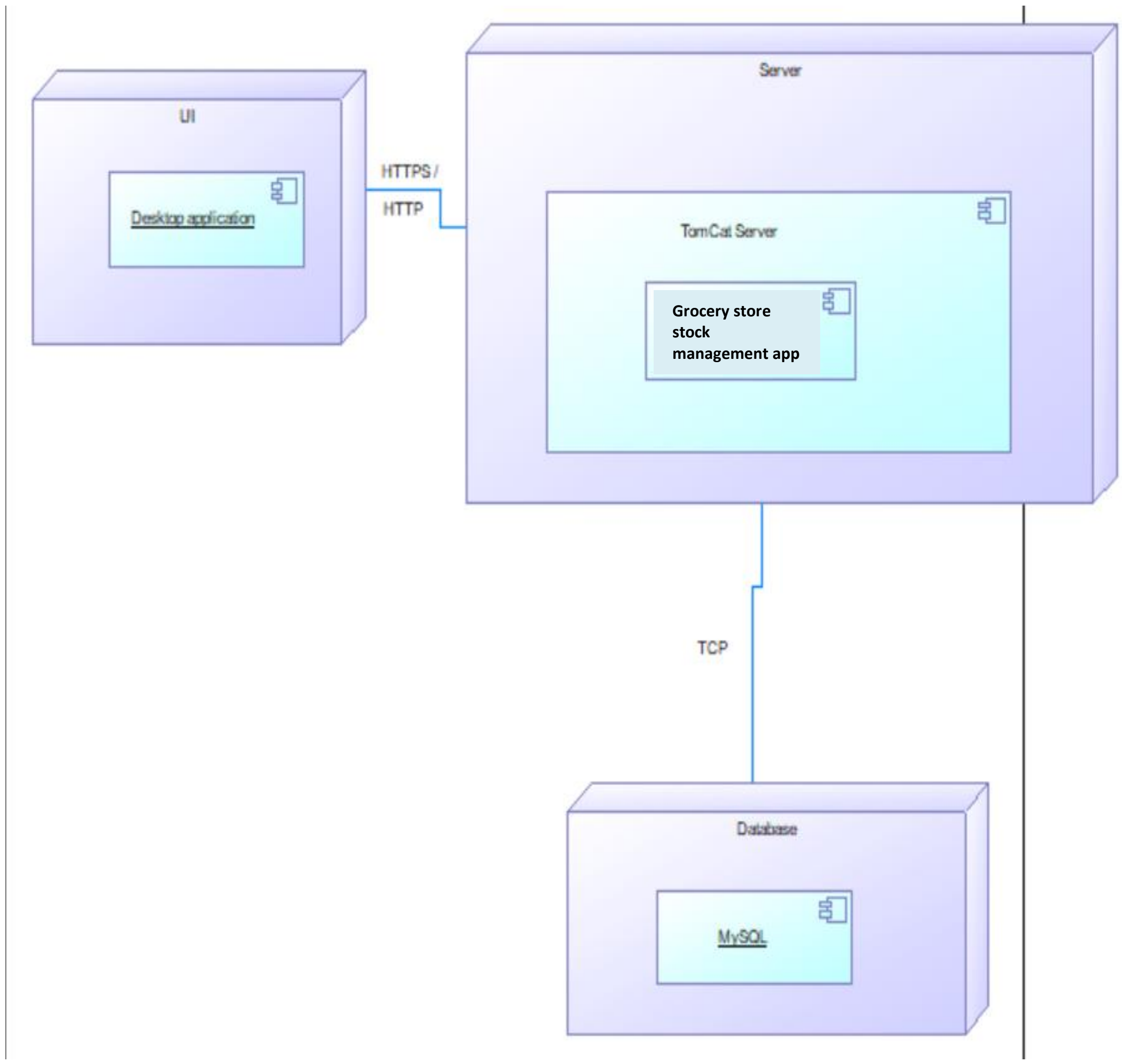
a) Definition

A deployment diagram is a structural diagram that illustrates the physical components and relationships within a system. It showcases the topology of where the software components are deployed. This diagram includes nodes, which represent the physical hardware used to deploy and run the application.

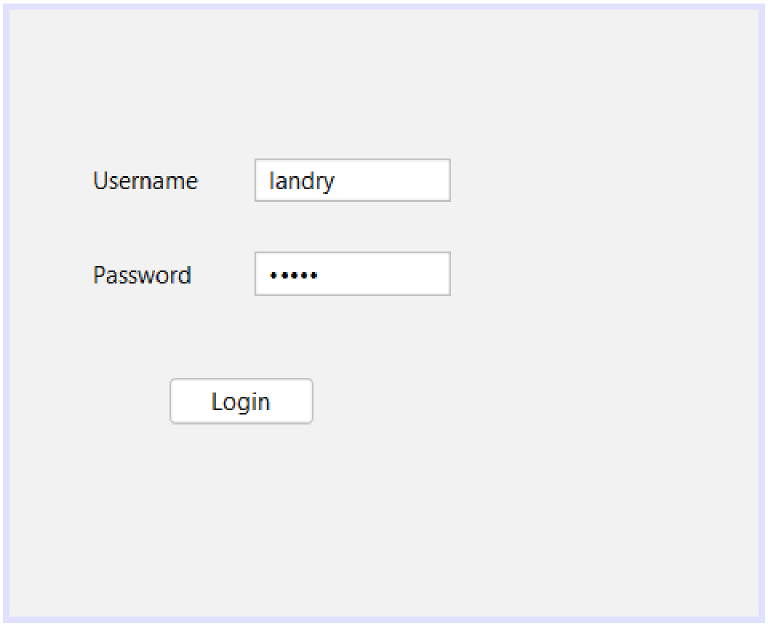
b) Deployment Diagram Components

Element	Description and main properties	Notation
Node	It is a hardware used to deploy the application.	
Artifact	An artifact is a major product, which is produced or used during the development of a software. E.g., diagrams, data models, setup scripts	
Component	It represents a modular part of a system that encapsulates its content and whose manifestation is replaceable within it environment.	
Association	An association helps to connect two nodes together which permits them to communicate together.	

c) System Deployment Diagram

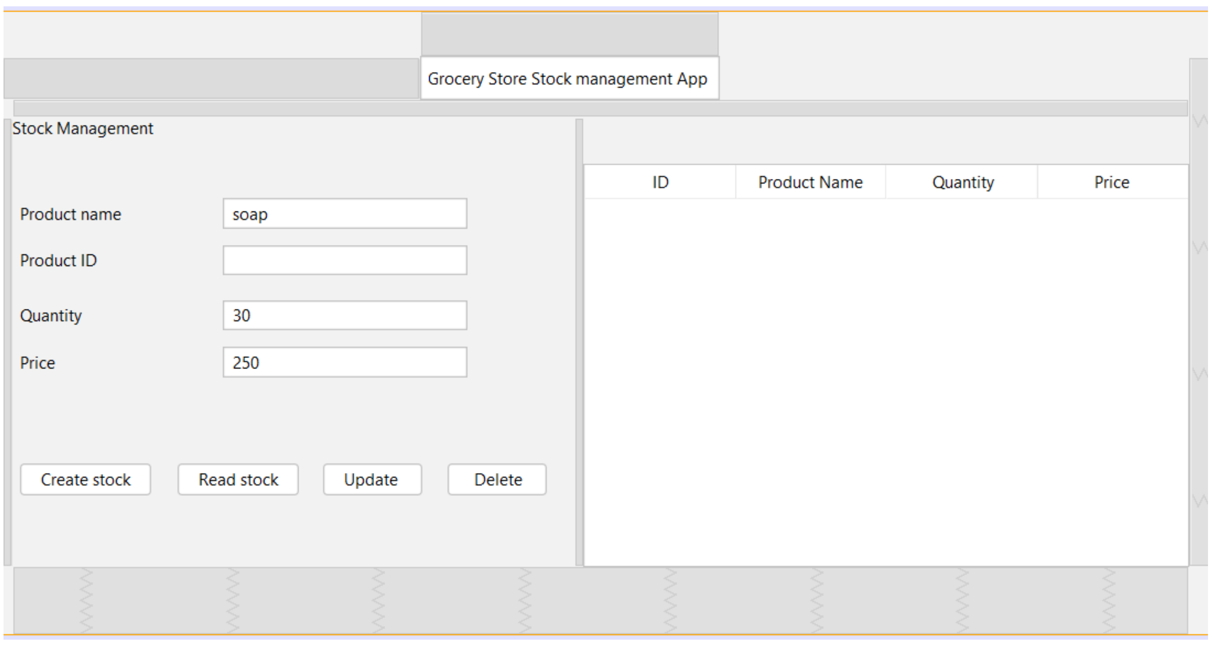

UI
Desktop application
HTTPS / HTTP
Server
TomCat Server
Grocery store stock management app
TCP
Database
MySQL

IV – CAPTURES OF THE APP



Username landry

Password •••••



Grocery Store Stock management App

Stock Management

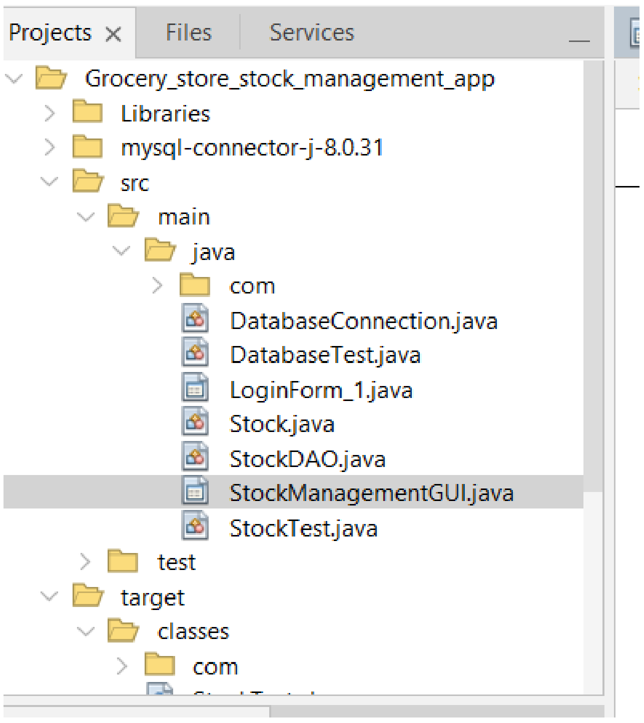
Product name soap

Product ID

Quantity 30

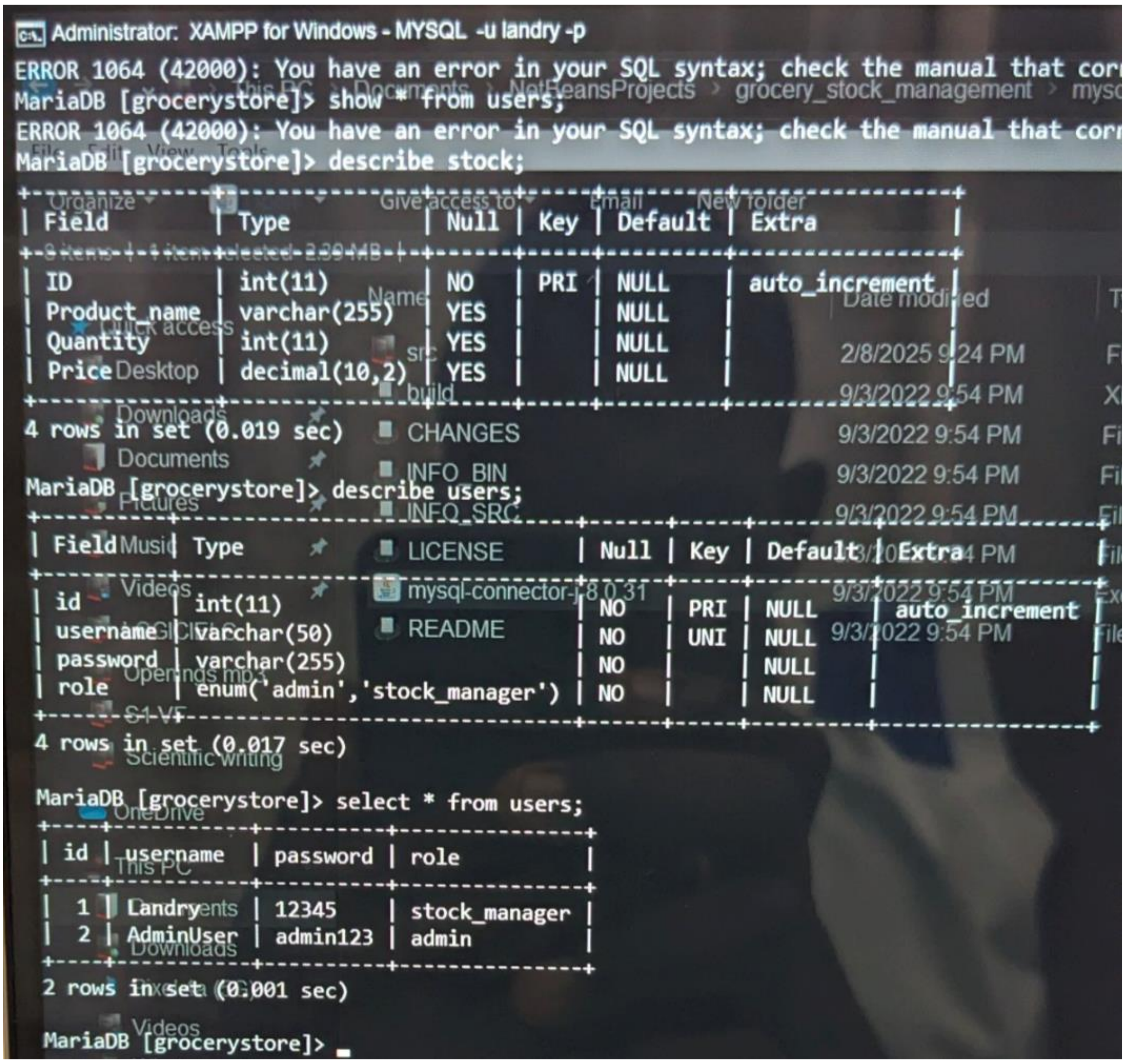
Price 250

ID	Product Name	Quantity	Price



Projects × Files Services

```
Grocery_store_stock_management_app
  Libraries
  mysql-connector-j-8.0.31
  src
    main
      java
        com
        DatabaseConnection.java
        DatabaseTest.java
        LoginForm_1.java
        Stock.java
        StockDAO.java
        StockManagementGUI.java
        StockTest.java
    test
  target
    classes
      com
```



```
Administrator: XAMPP for Windows - MYSQL -u landry -p
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that cor
MariaDB [grocerystore]> show * from users;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that cor
MariaDB [grocerystore]> describe stock;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| ID           | int(11)       | NO   | PRI | NULL    | auto_increment |
| Product_name | varchar(255)  | YES  |     | NULL    |                |
| Quantity     | int(11)       | YES  |     | NULL    |                |
| Price        | decimal(10,2) | YES  |     | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+
4 rows in set (0.019 sec)

MariaDB [grocerystore]> describe users;
+----------+--------------------------------+------+-----+---------+----------------+
| Field    | Type                           | Null | Key | Default | Extra          |
+----------+--------------------------------+------+-----+---------+----------------+
| id       | int(11)                        | NO   | PRI | NULL    | auto_increment |
| username | varchar(50)                    | NO   | UNI | NULL    |                |
| password | varchar(255)                   | NO   |     | NULL    |                |
| role     | enum('admin','stock_manager')  | NO   |     | NULL    |                |
+----------+--------------------------------+------+-----+---------+----------------+
4 rows in set (0.017 sec)

MariaDB [grocerystore]> select * from users;
+----+-----------+----------+---------------+
| id | username  | password | role          |
+----+-----------+----------+---------------+
|  1 | Landry    | 12345    | stock_manager |
|  2 | AdminUser | admin123 | admin         |
+----+-----------+----------+---------------+
2 rows in set (0.001 sec)

MariaDB [grocerystore]> _
```

V - CONCLUSION

The Grocery Store Stock Management System successfully meets its objectives by providing an efficient way to manage inventory and user authentication. The project demonstrates the practical application of Java and MySQL in a real-world inventory management scenario.