BSSE SEMESTER PROJECT

Computer Science Department

2024

Project name:

Point of sell for marble factory

Group member:

- *M Umair Latif (223644)* 6.5 / 10
- *Rao Ali Ahmed (223710)* 6.5 /10

Supervisor Name:

PROF. UBAID BIN ZAFAR

→ Major mistakes in diagrams. (suggested) (must be rectified).

→ wrong choosen architecture.

Product description:

The Marble Factory POS system is a comprehensive, user-friendly solution designed to streamline sales, inventory management, billing, reporting, and invoicing for a marble factory. Built on a Layered Architecture, it ensures robust performance, scalability, and ease of maintenance, providing real-time data and accurate transaction handling through intuitive interfaces for both users and administrators.

Functional requirements:

1. The system shall Print bill.
2. The system shall generate report & monthly profit.
3. The system shall generate report of remaining stock of goods.
4. The system shall give rights to the Admin to change Price and inventory products.

Non-Functional requirements:

1. The system should be run on Linux, window.
2. The system should run on multi-PC.
3. The system should must sure the safety of customer data
4. The system should print the bill in one second.
5. the system should print bill in medium size.

Domain requirements:

1. Ability to input and manage various marble product types including slabs. Liles, and custom. order. store product descriptions, images and pricing information

2. Real-time tracking of marble Stock level Integration with suppliers for reordering raw material

3. Calculate order total including taxes. and Profit.

4. Capture customer orders, including product Selection, quantities and delivery details.

Uml diagram

We can elaborate our requirements in system design, here the following uml diagram

1.use case
2.class diagram
3.Activity diagram
4.steamline diagram
5.sequence diagram

Use case:

A use case diagram is a visual representation of the interactions between users (or "actors") and a system, showcasing the various ways they use the system. It's a part of the Unified Modeling Language (UML) and is primarily used to capture the functional requirements of a system.

Scenario:

Use case for point of sell (pos) for marble factory. The system should allow the user/admin to login, print bill, check inventory, monthly report, check invoices. Also inventory manager manages all the inventory products.

Actors:

- Admin
- Inventory manager

Pre -condition:

1. The system is on, run by the admin/user.

Post -condition:

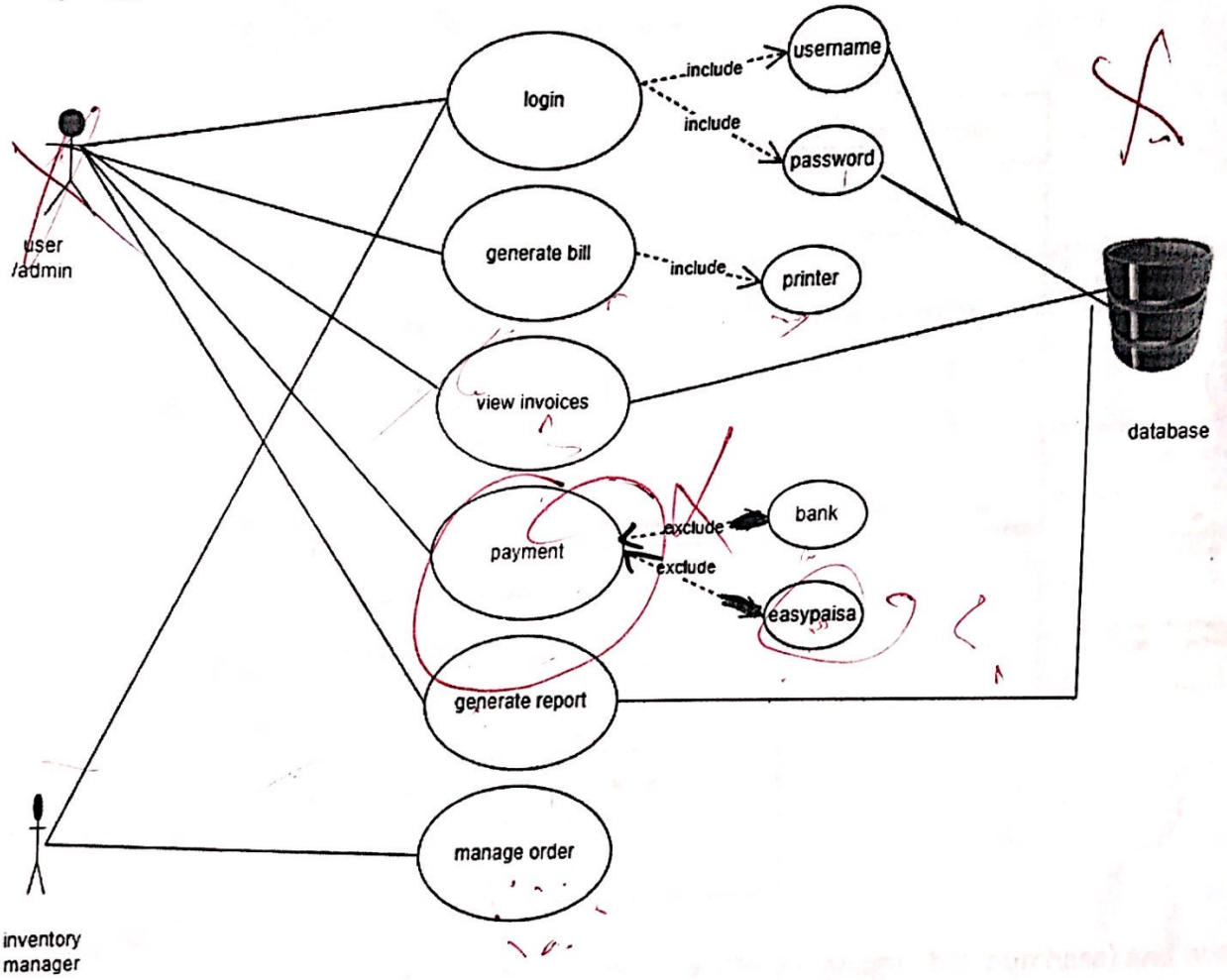
1. The system updates all the records.

Events of flow:

1. User/Admin/inventory manager enters credentials and clicks 'Login'.
2. System validates credentials.
   - Valid: Grants access and logs the event.
   - Invalid: Displays error message.
3. User/Admin completes a transaction.
4. System generates and sends the bill to the printer.
5. User/Admin selects 'Check Inventory'.
6. System retrieves and displays inventory status.
7. User/Admin selects 'Generate Monthly Report'.
8. System compiles and generates the report.
9. Report is displayed/printed.
10. User/Admin selects 'Check Invoices'.
11. System retrieves and displays invoices.

12. User/Admin manages invoices.
13. Inventory Manager selects 'Inventory Management'.
14. System retrieves and displays inventory data.
15. Inventory Manager manages inventory.
16. Changes are saved and event logged.

Diagram:



Class diagram:

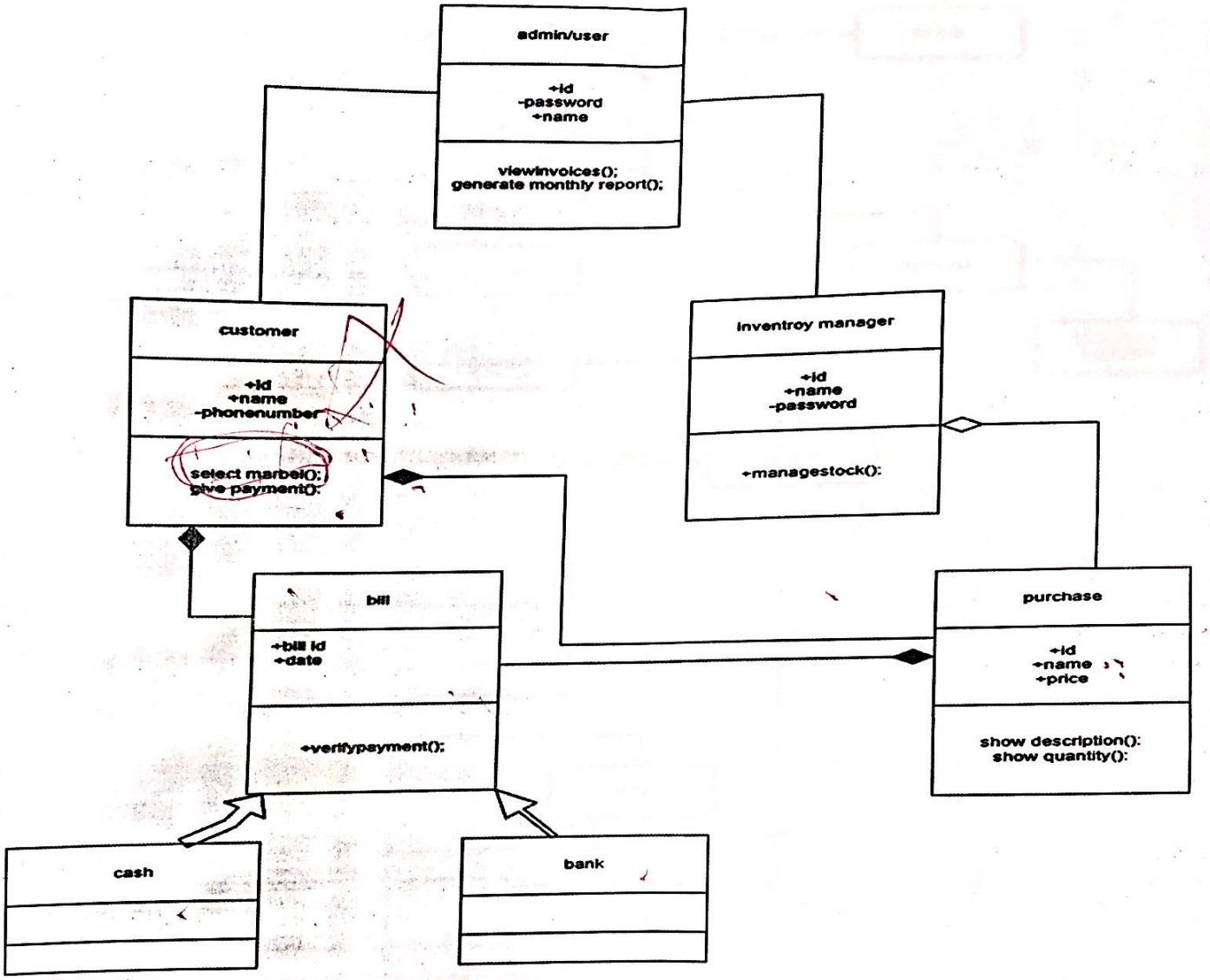
A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Scenario:

The class diagram defines the structure of the POS system for a marble factory, focusing on user roles, transactions, billing, inventory management, reporting,

and invoice management. It illustrates how various objects interact to provide the required functionalities and maintain system integrity.

Diagram:



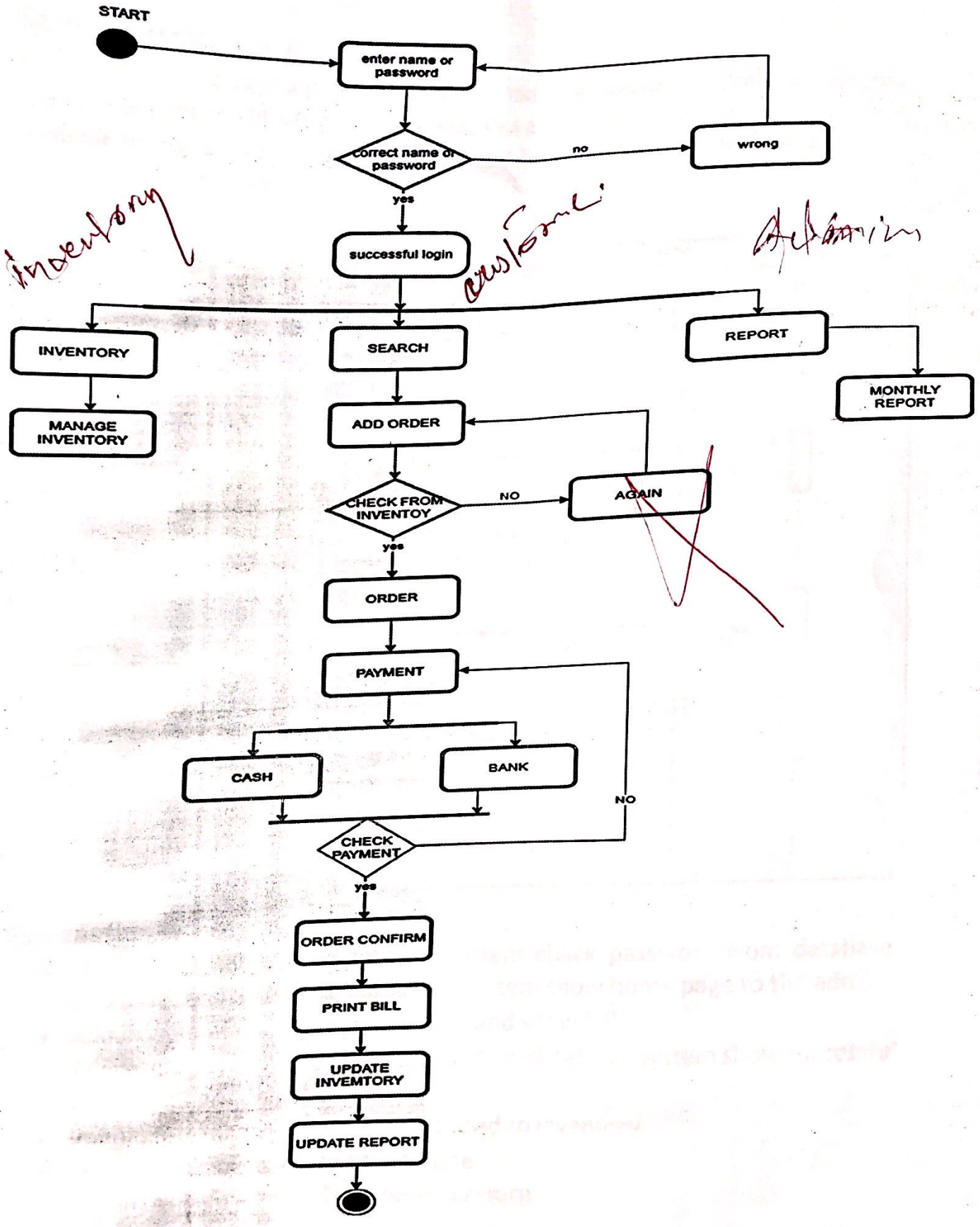
Explanation:

We made four main class which Is (admin, customer, inventory manager, bill, purchase) and two child classes (cash, bank). We use the relation of parent child, aggregation, composition.

Activity diagram:

An activity diagram is essentially a flowchart that shows activities performed by a system.

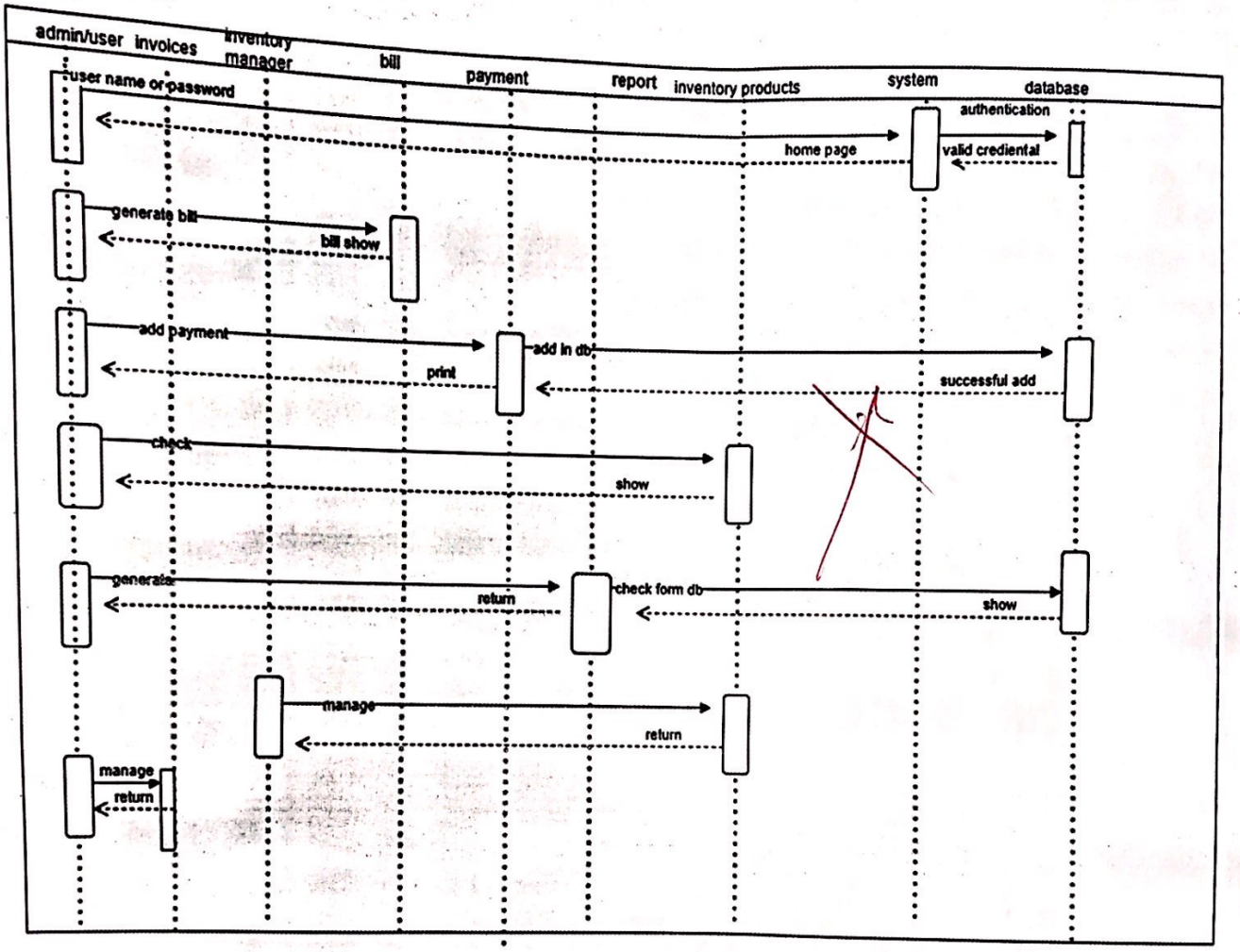
Diagram:


START
enter name or password
correct name or password
no
wrong
yes
successful login
INVENTORY
MANAGE INVENTORY
SEARCH
ADD ORDER
CHECK FROM INVENTOY
NO
AGAIN
yes
ORDER
PAYMENT
CASH
BANK
NO
CHECK PAYMENT
yes
ORDER CONFIRM
PRINT BILL
UPDATE INVEMTORY
UPDATE REPORT
REPORT
MONTHLY REPORT

Sequence diagram:

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

Diagram:



Explanation:

- User/admin enter the password. System check password from database. database return valid credential and system show home page to the admin.
- Admin generate bill, system calculate and show bill.
- Admin add payment then system update in database, system show successful update and print bill.
- Inventory manager all the record related to inventory.
- Admin check inventory from database.
- Admin manage invoices and invoices return.

Architecture

For a Point of Sale (POS) system for a marble factory with the specified requirements, the Layered Architecture is the best fit. This architecture provides a structured approach, separating concerns into different layers, which simplifies development, maintenance, and scalability.

Layered Architecture:

The Layered Architecture organizes the system into distinct layers, each with a specific responsibility. Typically, these layers include the Presentation Layer, Business Logic Layer, Data Access Layer, and Database Layer. This Layered Architecture provides a clear separation of concerns, with each layer focusing on a specific aspect of the application. The Presentation Layer handles the user interface, the Business Logic Layer processes requests and applies business rules, the Data Access Layer interacts with the database, and the Database Layer stores and manages the data. This structure enhances modularity, maintainability, and scalability, making it ideal for the POS system for a marble factory.

Diagram: