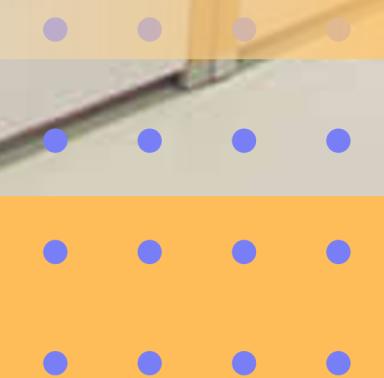




**LJ University**  
University with a Difference

# ELECTRONICS STORE MANAGEMENT

Optimize Your Store, Maximize Your Profits.





# Group Introduction



Dharmit  
Kakadiya  
Roll no.: - 104



Rushvi  
Doshi  
Roll no.: - 109



Shivangi  
Gohel  
Roll no.: - 110



Bhavesh  
Velani  
Roll no.: - 113

Enrollment  
No. :-

23002170110059

23002170110037

23002170210034

23002170210134



# Project Outline



- |           |                    |           |                       |
|-----------|--------------------|-----------|-----------------------|
| <b>01</b> | Purpose of project | <b>07</b> | Ds and Class Used     |
| <b>02</b> | Project Overview   | <b>08</b> | Procedure and Trigger |
| <b>03</b> | ER Diagram         | <b>09</b> | Tools Used            |
| <b>04</b> | Use case Diagram   | <b>10</b> | Merits                |
| <b>05</b> | Functionalities    | <b>11</b> | Demerits              |
| <b>06</b> | Implementation     | <b>12</b> | References            |

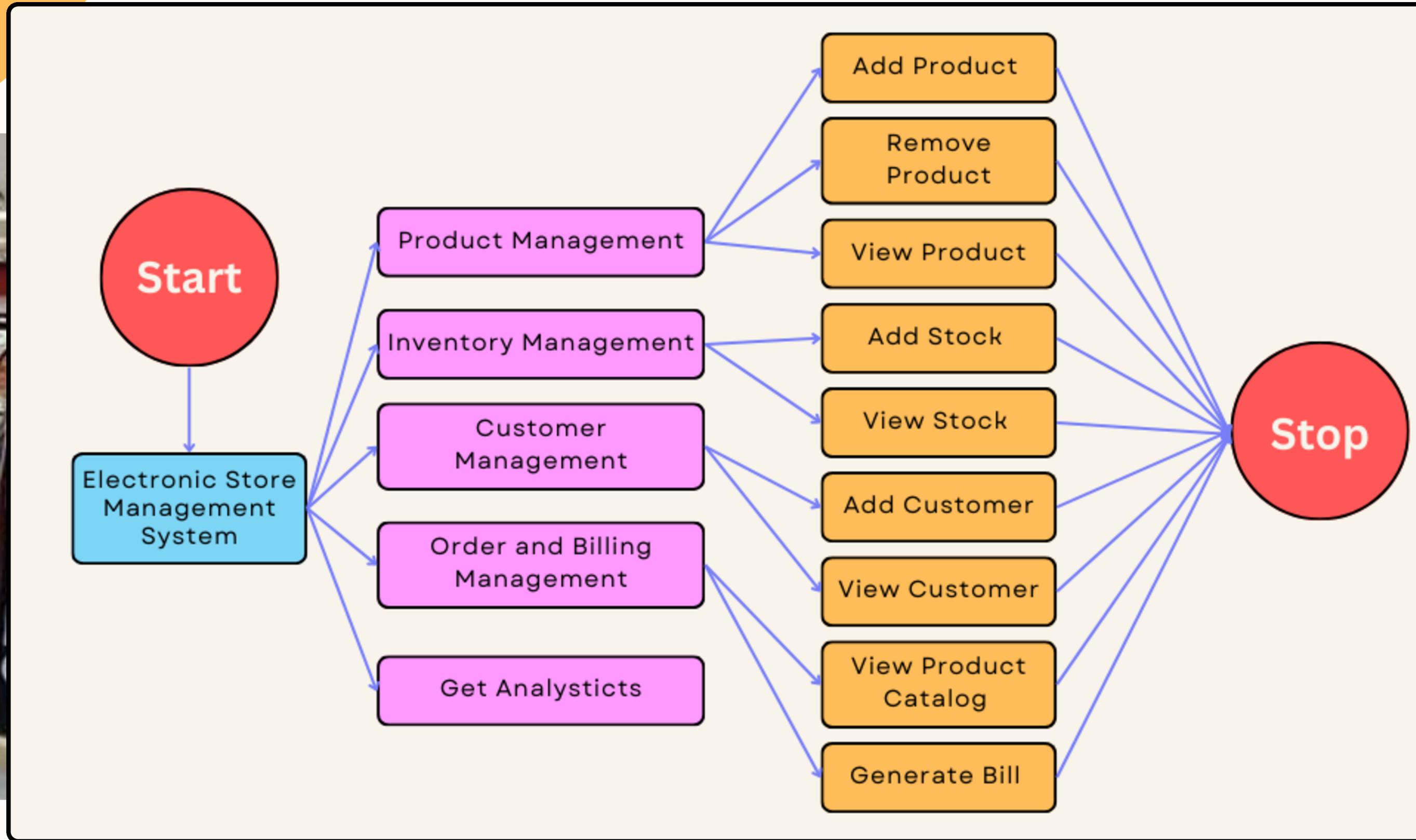
# Purpose of project



- **Product Management**
- **Inventory Management**
- **Customer Management**
- **Order and Billing**

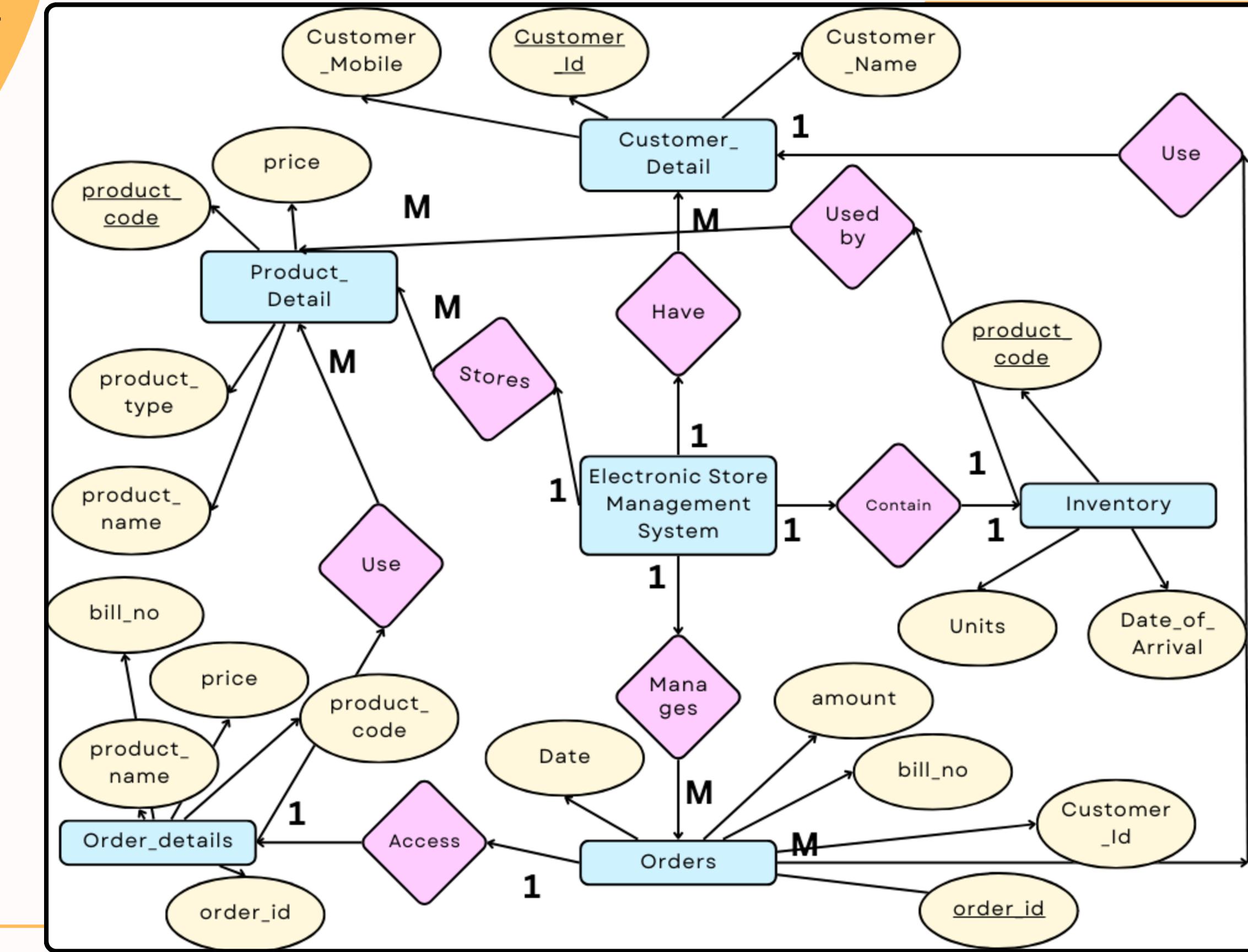
## 2

# Project Overview



# 3

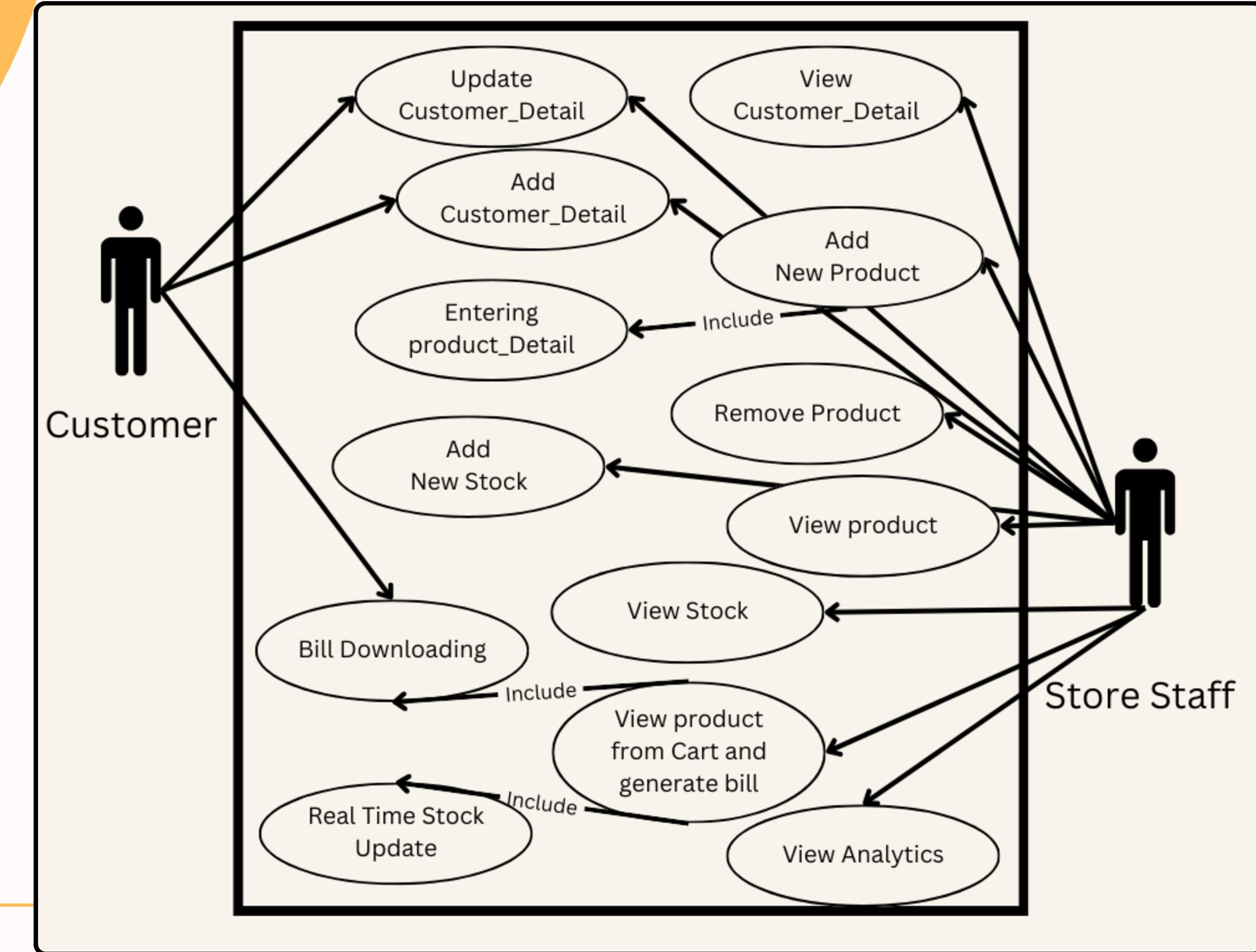
# ER Diagram





4

# Usecase Diagram

LJ University  
University with a Difference

## 5

# Functionalities



- Real Time Stock update
- Stores Order History
- Bill Downloading
- Get Analysis
- Manages Customer Details



# 6

# Implementation

- For DS -
  1. **Linked List**
    - Add
    - Display
  2. **ArrayList**
  3. **Priority Queue**

- For DBMS -
  1. **Procedure**
  2. **Trigger**
  3. **SubQueries**
  4. **DBMS basics**

- For JAVA -
  1. **OOPs**
  2. **File I/O**
  3. **JDBC**
  4. **Packages**



**LJ University**  
University with a Difference



# 7

# Ds and Class used



- **LinkedList Used for Customer Class**

- Insert Node

```
/* Method to add a new node with  
given data at the end of the list*/  
public void addLast(Customer p) {  
    Node newNode = new Node(p);  
    if (head == null) {  
        head = newNode;  
    } else {  
        Node current = head;  
        while (current.next != null) {  
            current = current.next;  
        }  
        current.next = newNode;  
    }  
}
```



- Display details

```
// Method to display all details of customer class  
public void display() {  
    if (head == null) {  
        System.out.println("List is empty");  
        return;  
    }  
    Node temp = head;  
    while (temp.next != null) {  
        temp = temp.next;  
        System.out.println("Customer Code: " + temp.p.c_id);  
        System.out.println("Customer Name: " + temp.p.c_name);  
        System.out.println("Customer Mobile No. : " + temp.p.c_mobile);  
        System.out.println();  
    }  
}
```

- **ArrayList used for Order class**
- **Priority Queue used for Product class**



# 8

# Procedures and Trigger



- CheckCustomerId

Trigger name: CheckCustomerId

Table: customer\_details

Time: BEFORE

Event: INSERT

```
1 BEGIN
2   IF EXISTS (SELECT c_id FROM Customer_details WHERE c_id
= NEW.c_id) THEN
3     SIGNAL SQLSTATE '45000'
4     SET MESSAGE_TEXT = 'Duplicate CustomerId is not
allowed.';
5   END IF;
6 END
```

- Trigger Message

MySQL said:

#1644 - Duplicate CustomerId is not allowed.

- getAnalytic

Routine name: getAnalytics

Type: PROCEDURE

Direction	Name	Type	Length/Values
IN	p1_code	INT	
OUT	p1_type	VARCHAR	50
OUT	p1_price	DOUBLE	

Add parameter

```
1 BEGIN
2 SELECT p_type,p_price into p1_type,p1_price from product_details where p_code = p1_code;
3 END
```

- StockUpdate

Details

Routine name: StockUpdate

Type: PROCEDURE

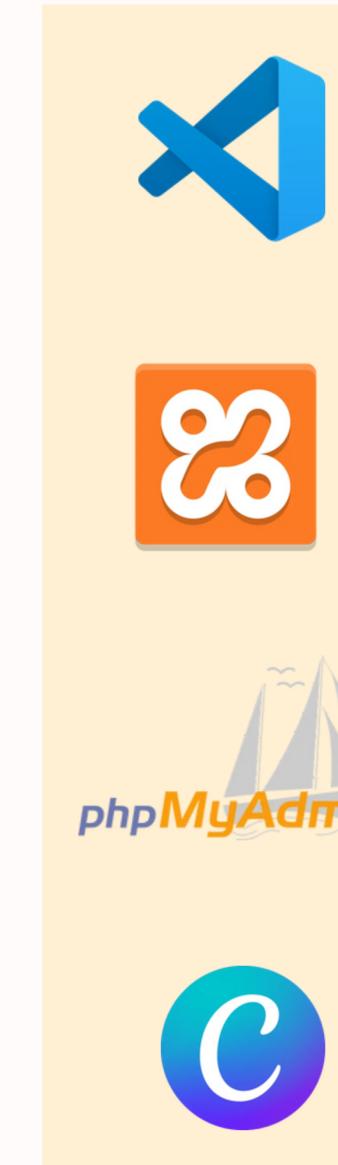
Direction	Name	Type	Length/Values
IN	p1_code	INT	

Add parameter

```
1 BEGIN
2 UPDATE inventory set units = (units-1) where p_code= p1_code;
3 END
```



9



# Tools used

**Visual Code Studio**

**Xampp Control Panel**

**phpMyAdmin**

**Canva**

## 10

# Merits



- **Real-Time Tracking** : Allows for real-time tracking of stock levels, helping prevent overstocking or stockouts.
- **Revenue Analysis** : Provides detailed reports on sales, profits helping with budgeting and financial planning.
- **Operational Efficiency** : Automates routine tasks such as billing, invoicing, and stock management, reducing manual effort and the risk of errors.

# 11

## Demerits



- **System Downtime:** Technical issues or system failures can disrupt store operations and affect sales and customer service.
- **Initial Investment:** Setting up a store management system can involve significant upfront costs for software, hardware, and installation.
- **Software Upgrades:** Updates to the software may be necessary to address bugs or add new features, which could require additional investment.



# References

- ChatGPT
- Chrome
- Faculties:
  1. Parth Thakkar
  2. Pooja Vyas
  3. Divyang Patel



**LJ University**  
University with a Difference

# THANK YOU

From Our Group Members



