Requirements Analysis
A report on Analytical Information System for Tesco drive thru

1. Introduction:

Tesco was founded by Jack Cohen in 1919. It is one of the biggest retailers in the world, with its main office located in Welwyn Garden City, Hertfordshire, England

1.1 What TESCO does:

Tesco serves a wide spectrum of consumer demands, from big hypermarkets to neighbourhood convenience stores, with a variety of store formats that include Tesco Superstores, Tesco Extra, Tesco Express, and One Stop shops.

Tesco is a grocery, apparel, electronics, and financial services retailer with a wide selection of products. It places a strong emphasis on both branded and own-label products. With delivery and click-and-collect options, Tesco.com, its strong online platform, improves customer convenience.

Tesco's Clubcard loyalty program cultivates enduring ties with its consumers by offering prizes. Tesco is dedicated to sustainability and actively seeks to cut down on carbon emissions, plastic consumption, and food waste. In addition to having a large economic influence, Tesco, a FTSE 100 firm and major employer, participates in charitable work and community projects, supporting food banks and educational initiatives.

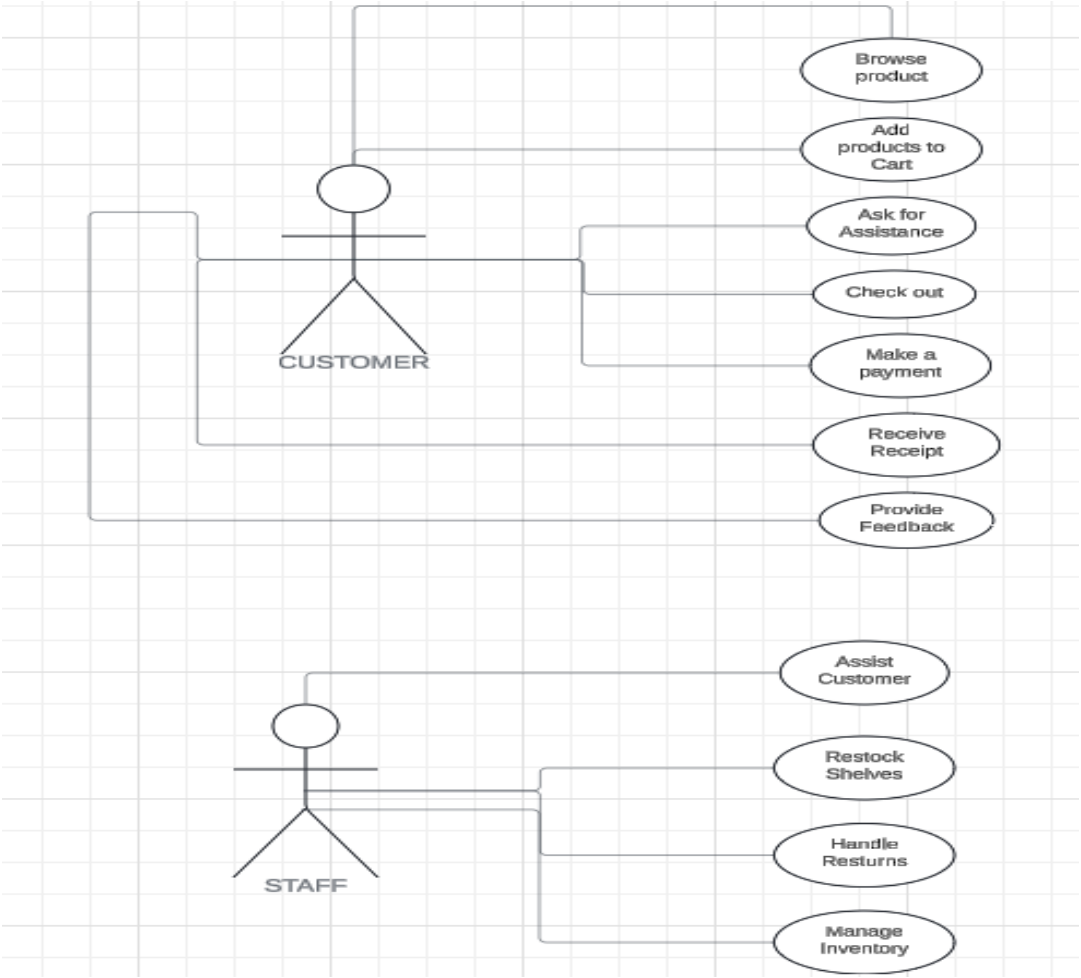
1.2 The reasons why TESCO needs new analytics capabilities:

Tesco can benefit from a number of important features and benefits when they implement a drive-thru concept, which will increase consumer convenience and operational efficiency. This system's most notable feature is its simplified online ordering interface, which can be accessed through both web and mobile applications.

Customers can use this platform to order products, browse the product catalog, and check the inventory in real time while at home or on the road. Customers can choose their preferred pickup times with the scheduling feature, which helps control traffic and shorten drive-thru wait times.

Tesco will have an advantage over rivals thanks to the drive-thru concept, which offers a unique service to satisfy customers' increasing need for contactless shopping choices. By providing a quick, easy, and secure buying experience, it can draw in new clients and boost existing ones. Tesco is able to optimize marketing campaigns, make data-driven decisions, and customize offerings to fit customer needs because to the system's ability to gather and analyse data on customer activity and order patterns. In the end, this creative strategy strengthens Tesco's will to adjust to shifting consumer tastes and industry trends while also increasing sales and operational effectiveness.

2.Operational Data Scheme:

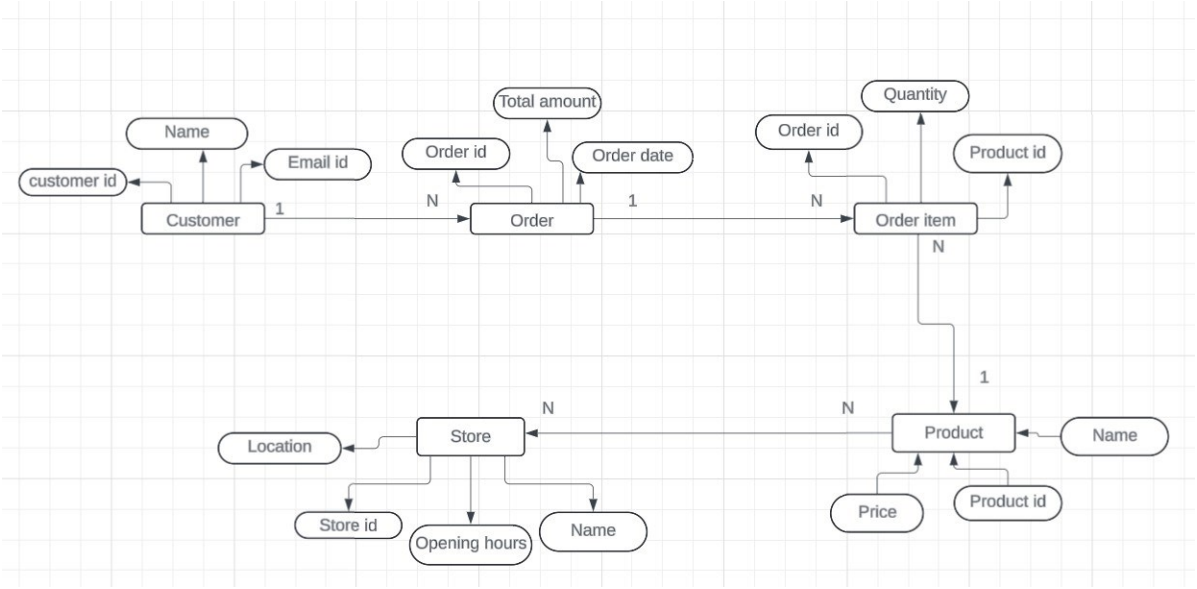


2.1 Analytical information system Use case Diagram

OBSERVATION:

- **Customer Interactions:** The major tasks that customers perform—browsing products, adding items to their basket, requesting assistance, checking out, paying for their purchases, obtaining receipts, and giving feedback—are succinctly depicted in the diagram.
  These use examples emphasize the significance of customer service and operational efficiency while highlighting the key touchpoints in the customer buying experience.
- **Staff Interactions:** The staff's responsibilities are clearly defined, with a focus on managing inventory, assisting customers, replenishing shelves, and processing returns. These use cases highlight the vital support tasks employees carry out to guarantee clients have a flawless purchasing experience.

Conclusion: The present use case diagram provides a strong basis for comprehending the fundamental procedures and interactions in TESCO's physical retail setting. The diagram has the potential to become a comprehensive tool for analysing and managing retail operations, leading to improvements in customer happiness and operational efficiency. This can be achieved by adding more actors and providing more details about the use cases.



2.2 ER DIAGRAM

ERD is an actual representation of an organization's working data and contains entities like Customer, Order, Order item, Store and Product in the case of Tesco Supermarket.

Consumers place orders with several order items, each of which relates to a different product. A key component of inventory management is the many-to-many connectivity between products that are stocked across several locations. A clear understanding of these connections promotes efficient data management and seamless retail operations.

REALTIONSHIPS:

- Customer to Order - One to Many relationships, where a customer can place multiple orders.
- Order to Order Item - One to Many relationships, Multiple order items may be included in a single order, but each order item is associated with a single order.
- Order Item to Product - Many to One relationship, each order item references one product.

- Stores stocks Product – Many to many relationships, A store can stock multiple products, and a product can be stocked in multiple stores.

2.3 Structured Query Language (SQL)

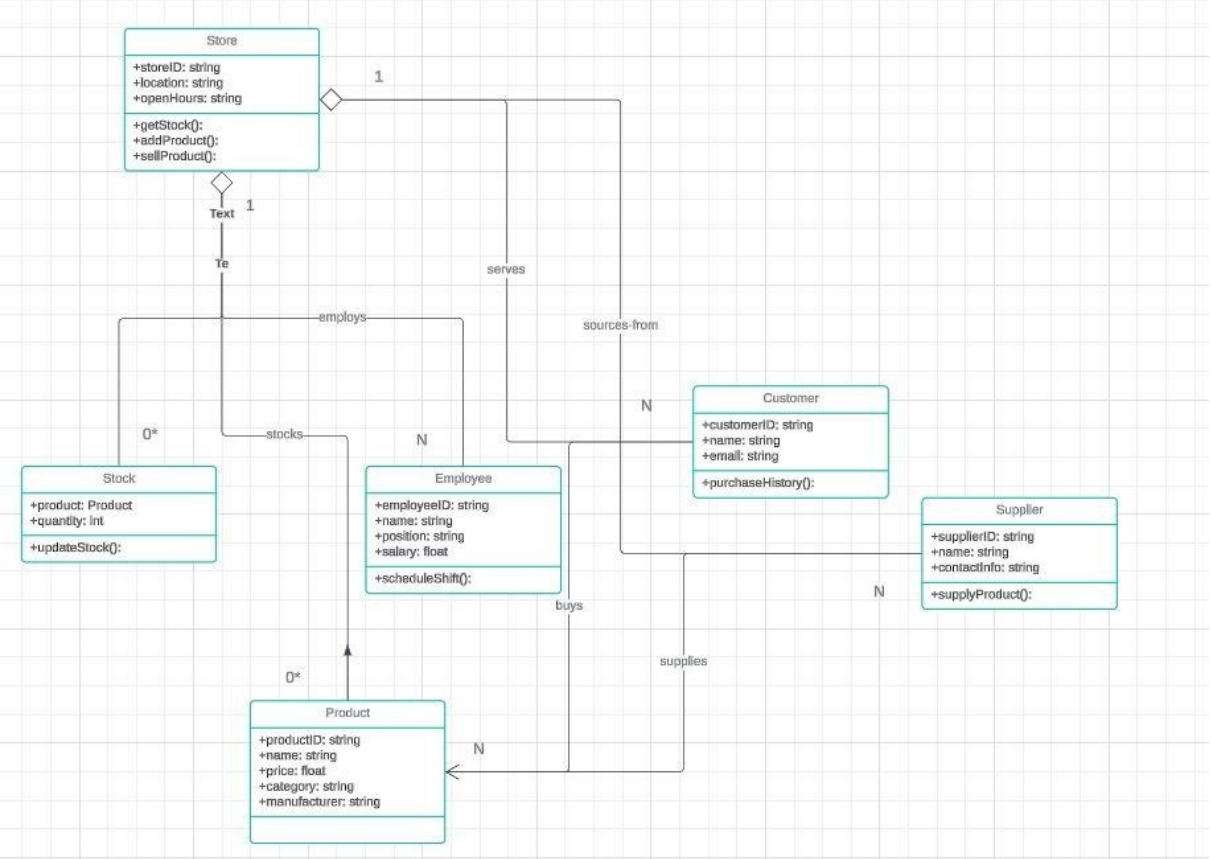
```
CREATE TABLE Customers (
Customer id INT PRIMARY KEY,
Name VARCHAR (100),
Email id VARCHAR (100),
);

CREATE TABLE Orders (
Order id INT PRIMARY KEY,
Customer ID INT,
Orderdate Date,
Totalamount Decimal (10,2),
FOREIGN KEY (Customer Id) References Customers (Customer Id)
);
CREATE TABLE Products (
ProductId INT PRIMARY KEY,
Name VARCHAR (100),
Price Decimal (10,2)
);

CREATE TABLE Order items (
Order itemid INT PRIMARY KEY,
Order id INT,
ProductId INT,
Quantity INT,
);
CREATE TABLE STORES (
Store Id INT PRIMARY KEY,
Name VARCHAR (100),
```

Location VARCHAR (100),

Opening hours VARCHAR (50),

);

2.4 UML DIAGRAM



OBSERVATION:

1. **Store employs Employee:** One to Many
   Multiple employees may be employed by a single business (identified by storeID), but each employee (identified by employeeID) is only employed by one store.

1. **Store Stocks Product (via Stock):** Many to Many
   Products can be stocked at many stores by each store, and multiple stores can stock the same product. The Stock entity, which uses storeID and productID as foreign keys, is in charge of managing this relationship.

2. **Store serves Customer:** One to Many
   Although a person can shop at many locations, each establishment serves multiple customers. Transaction records can be used to handle this relationship.

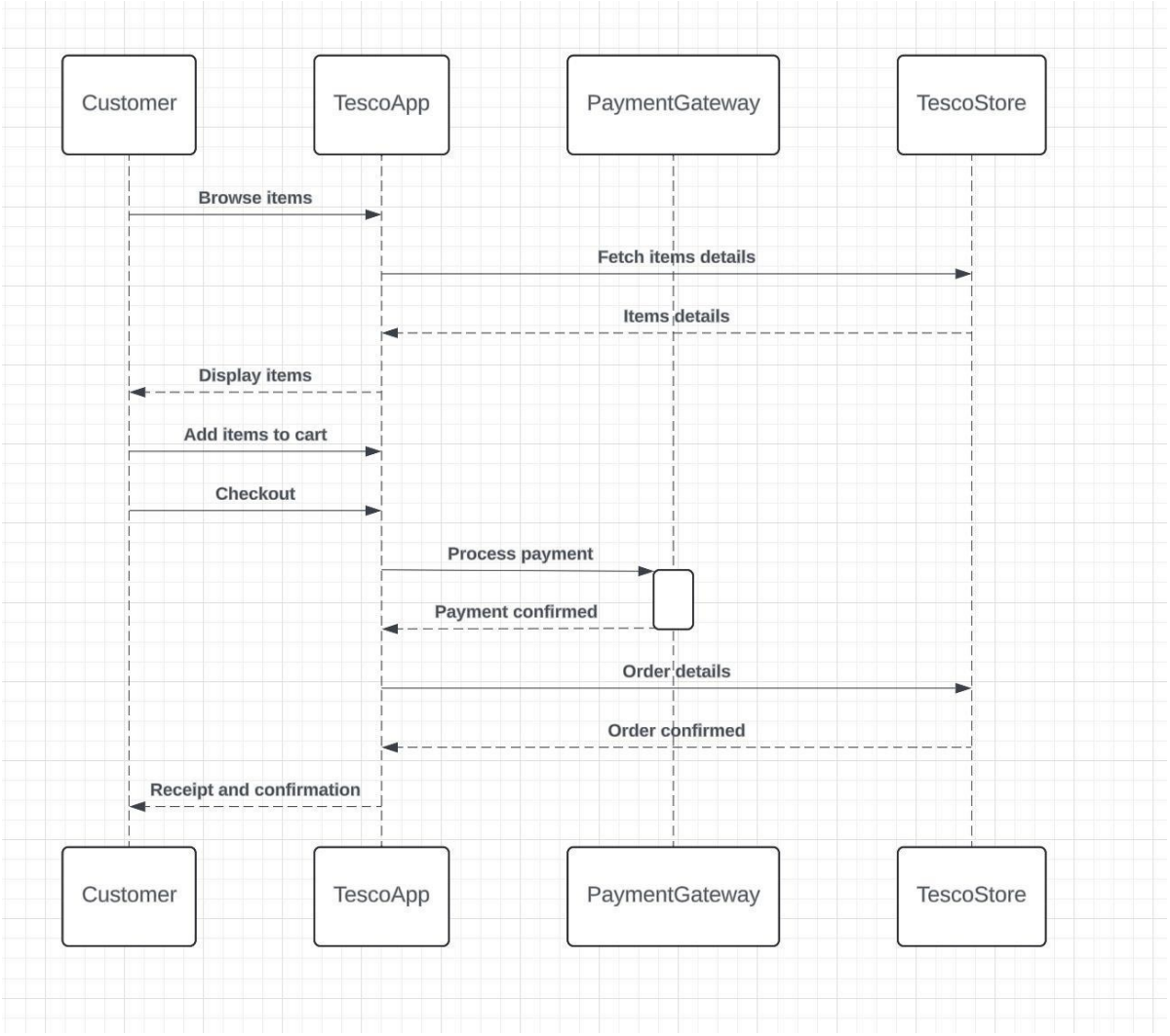
3. **Customer buys Product:** Many to Many
   Customers are able to purchase numerous items, and several customers may purchase each item. Usually, transaction records are used to handle this relationship.

4. **Stores sources- from supplier:** Many to Many
   Products can be sourced by each store from a variety of vendors, and suppliers can supply products to a number of retailers. Supply Record Table is used to maintain this relationship.

5. **Suppliers supplies product:** One to Many
   Though each product is often supplied by a single supplier, each supplier is able to supply several items. The supplierID in the Product object indicates this relationship.

CONCULSION: The key entities and their primary and foreign keys are shown in detail in the modified UML class diagram. This guarantees data integrity across many entities and improves the clarity of relationships. The graphic helps with effective management and operational workflows by showing how stores interact with staff, consumers, suppliers, and items. In order to ensure seamless and efficient retail operations, this comprehensive depiction helps to understand the structure of the system and the relationships between various components.

2.5 User Interaction Sequence Diagram



The sequence diagram offers a thorough rundown of the Tesco ecosystem's consumer purchase process, emphasizing the significance of smooth interactions between the payment processing systems, the store's order administration, and the customer-facing app. Tesco's objective of delivering an exceptional retail experience is in line with this organized flow, which guarantees accuracy, efficiency, and customer happiness.

3.ELICITATION:

3.1 Requirements Elicitation Plan

Techniques:

1. **Document Review and Stakeholder Analysis:**
   Identify stakeholders, define objectives, and develop initial requirements draft, this preparation phase is very essential to understand the business context and identify key participants and initial insights.

2. **Interviews:** In-depth insights into everyday operations, pain issues, and user needs are obtained through interviews with stakeholders, such as consumers, cashiers, and shop managers. Through direct communication, unique needs and expectations are identified, guaranteeing that the new technology solves problems in the real world and increases productivity overall. Interviews aid in the creation of a customized, user-friendly system that satisfies corporate goals and raises service standards by gathering thorough feedback.

3. **Surveys and questionnaires:** Surveys and questionnaires are used to collect quantitative data from a wide range of sources, such as suppliers and customers, in order to uncover common concerns, trends, and preferences. By effectively gathering a wide range of information, this strategy aids in prioritizing features and enhancements according to real user feedback. Through the utilization of these technologies, Tesco can guarantee that the improvements made to the system meet the demands of customers and operational requirements, resulting in increased efficiency and satisfaction.

4. **Workshops:** Workshops enable brainstorming and fine-tuning of system needs through collaborative meetings with key stakeholders, including managers, employees, and IT teams. This participatory method promotes a common understanding of objectives, prompts the resolution of disputes or ambiguities, and stimulates active engagement. Tesco can guarantee that the built system closely complies with stakeholder expectations and business goals by utilizing workshops, which will result in a more efficient and user-centered solution.

5. **Observation**: Directly seeing everyday activities is how observations are made in order to comprehend workflows, spot inefficiencies, and obtain current information. By ensuring that the system requirements are grounded in real user behaviors and operational challenges, this technique produces workable and efficient solutions.

6. **Prototyping:** Through early system models created through prototyping, suggested features can be seen and interacted with by stakeholders. This hands-on approach ensures that the finished solution effectively satisfies user needs and expectations by facilitating instant input and iterative refinement

Timeline:

Activity	Stakeholders	Date/Duration
Interviews	Executive Team, Operations manager, store managers	Week 1
Customer Surveys	Customers	Week 1-2
Questionnaires	Suppliers	Week 1-2
Workshops	Internal teams	Week 2
On-site observations	Store operations	Week 3
Document analysis	IT and Inventory Records	Week 3
Prototyping	All Stakeholders	Week 4-5

Overall approach: The general strategy Elicitation Plan uses a thorough, iterative method to efficiently collect specific needs. The first step in this strategy is to engage a wide range of stakeholders including internal and external Stakeholders.

We guarantee that a broad range of insights and viewpoints are obtained by combining elicitation approaches, such as surveys, observations, workshops, interviews, and document analysis, giving a comprehensive picture of the system's requirements. We can iteratively modify requirements and address any gaps or misunderstandings early on thanks to regular feedback and validation sessions.

4. Initial Requirements and Documentation:

4.1 Functional Requirement:

1. **Customer Management:**

   Registration: Permit users to register using their name, email address, phone number, and address.

   Authentication: Permit clients to safely sign in using their login information.

   Profile management: Permit users to examine and amend their personal data.

2. **Product Management:**

   Product Listing: Permit the catalog' s products to be added, updated, and removed.

   Product classification: Divide items into distinct categories to facilitate browsing.

   Inventory tracking: Monitor product stock levels instantly

.

3. **Order Management:**

   Order Placement: Permit clients to place product orders.

   Order tracking: Give clients the ability to monitor the progress of their orders.

   Order History: Make previous order history available to clients.

4. **Payment Processing:**

   Integration of Payment Gateways: Allow for the use of several payment options, such as PayPal, credit/debit cards, etc.

   Make sure that every payment transaction is handled securely.

   Creation of Receipts: Create and forward order receipts to clients.

5. **Feedback Management:**

   Gathering Customer input: Get client input on goods and services.

   Analyse client comments to raise the Caliber of your services.

   Allow consumers to rank goods and services using a rating system.

6. **Notification:**

   Order Status Updates: Notify clients with any changes to their order status.

   Send out promotional alerts to clients informing them of upcoming sales and deals.

   Remind clients about things they left in their shopping basket with cart reminders**.**

4.2 Non-Functional Requirement:

1. **Performance:**

   Response Time: Make sure that in two seconds, the system reacts to user input.

   Throughput: During peak hours in particular, efficiently handle a large number of transactions.

   Scalability: Expand the system's capacity to handle a rise in the quantity of users and transactions.

2. **Security:**

Data protection: Make sure that client information is secure and conforms with data protection laws (like the GDPR).

Strong authentication and permission procedures should be put in place to guard against unwanted access.

Encrypt and securely process all payment information to ensure safe and secure payment processing.

3. **Reliability:**

Aim for 99.9% uptime in terms of availability, and make sure the system is highly available.

Fault Tolerance: Put in place safeguards to prevent data loss and handle system failures gracefully.

Maintaining data accuracy and consistency throughout all transactions is known as data integrity.

4. **Usability:**

Create a user interface that is simple to use and intuitive to traverse.

Accessibility: Verify that the system complies with accessibility guidelines and is usable by people with impairments.

Consistent Experience: Ensure that the user experience is the same across various platforms and devices.

5. **Maintainability:**

Modular System Design: To make updates and maintenance simpler, use a modular system design.

Documentation: Give system administrators and developers access to thorough and understandable documentation.

Error Handling: To facilitate troubleshooting, put in place strong error handling and logging procedures.

6. **Portability:**

Multi-Platform Support: Make sure the system works on a number of platforms, such as mobile and the web.

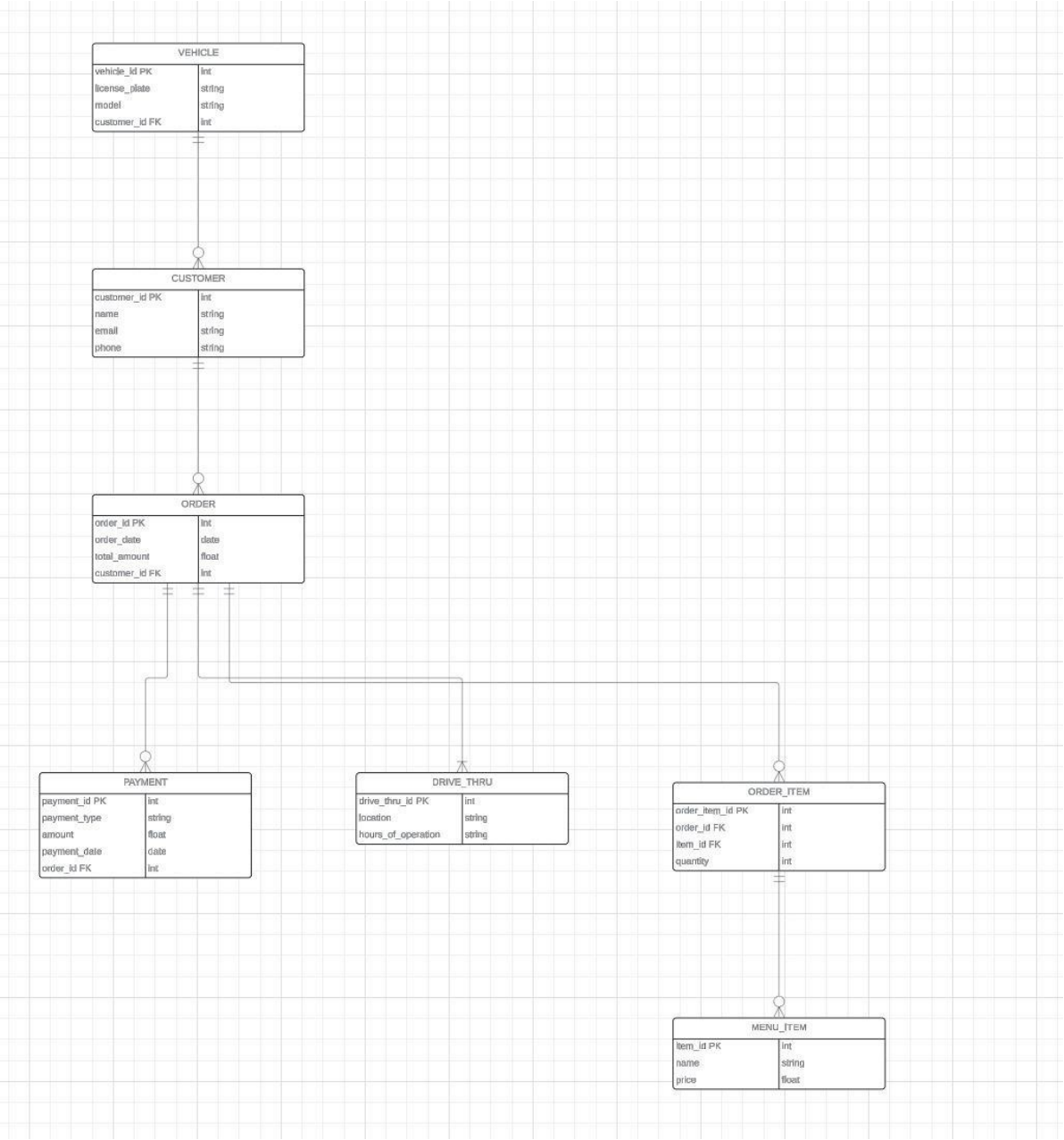
Verify compatibility with several web browsers to ensure browser compatibility.

Operating System Compatibility: The backend infrastructure should support a variety of operating systems

5. EHNANCEMENT:

5.1 UML after Enhancement

To infer a reasonable schema for TESCO DRIVE-THRU, we have to consider all the aspects of its operations including their services, customers, products, orders and payments.



Observation for Schema Design (UML):

1. **Vehicle –** The attributes are vehicle_id (PK) which is unique identifier for each vehicle, license plate, model and Customer_id (FK) which links to the customer entity.

2. **Customer –** Which captures the Tesco customers details attributes include customer_id, (PK), name, Email and Phone Every customer is guaranteed to be clearly recognized within the system thanks to their unique customer ID.

3. **Order –** Attributes Order_id (PK), Order date, Total amount and Customer_id (FK) This entity keeps track of customer orders, connecting each order to a unique customer.

4. **Payment** - This keeps track of order payments.

   The primary key, payment_id, payment_type, amount, payment_date, and order_id (a foreign key pointing to the Order object) are among the attributes. This guarantees that every payment is linked to a particular order, documenting the method, date, and amount of the payment.

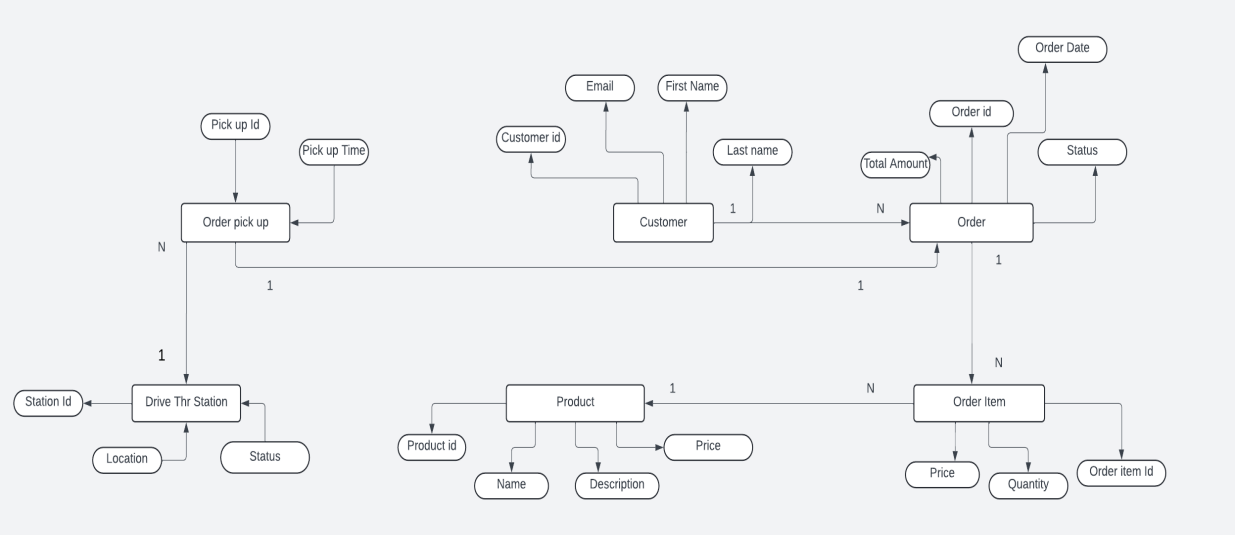
5. **Drive- Thru -** Information about the drive-thru stations is captured by the Drive Thru entity. The attributes consist of location, hours_of_operation, and drive_thru_id, which is the primary key. This organization makes sure the system can handle several drive-thrus, each with a unique ID and details about its location and hours of operation.

6. **Order Item -** This object lists the contents of every order. The attributes consist of the primary key order_item_id, a foreign key order_id that links to the Order entity, a foreign key item_id that links to the Menu Item entity, and quantity. This guarantees that every item inside an order is easily recognized and connected to the appropriate order and menu item.

7. **Menu Item -** The product information that are available for purchase are captured by the Menu Item entity. Name, price, and item_id—the primary key—are examples of attributes. This organization offers a library of products that clients can purchase, guaranteeing that every item is distinct from the others.

Conclusion: Tesco's drive-thru UML diagram incorporates a Vehicle entity, which links vehicles to customers, adding another level of information that can improve the order pick-up procedure. By making sure that every car is accurately identified and connected to the appropriate client and orders, this entity, along with its relationships to other entities, offers a more efficient and customized drive-thru experience. Tesco's drive-thru operations are effectively managed and consumers and operational personnel have a seamless experience because to the intricate relationships between customers, orders, payments, drive-thru stations, order items, menu items, and vehicles.

5.2 ER Diagram after Enhancement



OBSERVATION IN ER DIAGRAM

Customer:

Attributes: Customer id, First Name, Last Name and Email address

Observation: Customers' personal information is stored by the CUSTOMER entity. This contains the most basic contact information, which are necessary for locating and interacting with clients.

Order:

Attributes- Order-id, Order date, Order status and Total amount

Observation: Order-related data is gathered by the ORDER entity.

It contains information on the order, including the total amount, the date it was placed, and its current status. An association is created between the ORDER and CUSTOMER entities which links the order to the associated customer

Order Item:

Attributes- Order Item id, Quantity and Price

Observation: Each item that makes up an order is described in detail by the ORDER ITEM object.

The order item id, quantity, and price of each item are included in its record, together with information about the order to which it belongs. This organization links orders and items in an intermediary capacity.

Product:

Attributes- Product-id, Name, Description and Price

Observation: Product details are kept in the PRODUCT entity.

The product name, price, and stock level are examples of attributes. These are crucial details for order processing and inventory management. In ORDER ITEM, the product id attribute is utilized to associate order items with particular products.

Drive-Thru:

Attributes- Station ID, Location and Status

Observation: The organization guarantees the unique identification of every station and furnishes crucial operational information to facilitate the management of the station network. Every station's availability and operational state can be dynamically managed and communicated to clients by the system through the maintenance of a status characteristic.

Order Pick-Up:

Attributes- Pick Up ID, Pick Up Time.

Observations: An important part of organizing the last phase of the order fulfilment process is the Order Pick-Up entity.

Through precise capture of pick-up timings, purchases may be scheduled and delivered more efficiently, improving customer satisfaction and operational effectiveness overall.

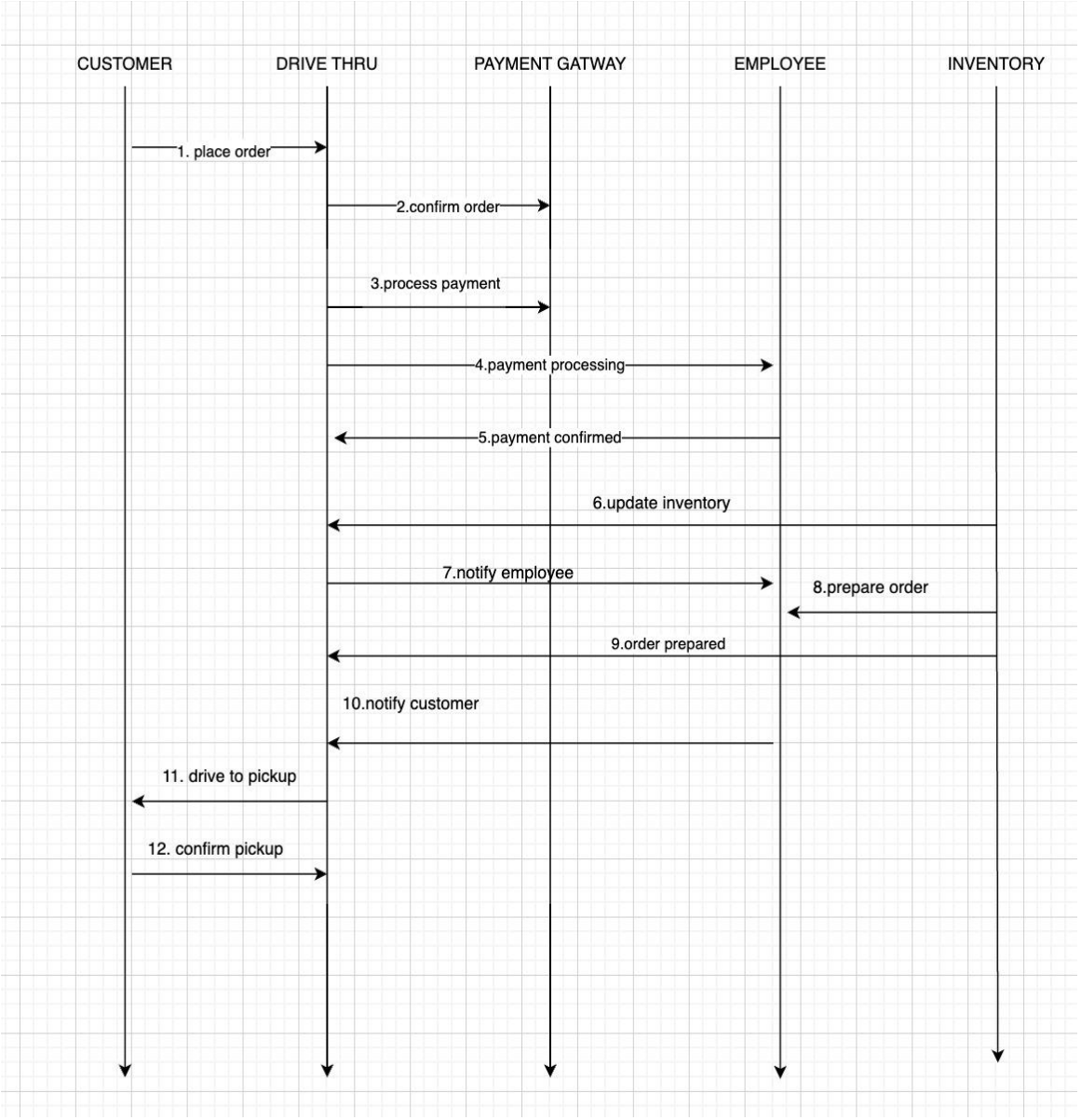
REALTIONSHIPS:

- Customer to Order - One to Many relationships, where a customer can place multiple orders.
- Order to Order Item - One to One or One to Many relationships, An order item represents each product that can be included in an order, however each order item is only a part of a single order.
- Order Item to Product - Many to One relationship, each order item references one product.
- Order to Order Pick-Up- One to One relationship, every order is linked to a single pick-up event since it gets picked up only once.
- Drive-Thru station to Order Pick up – One- Many relationships Multiple orders can be picked up at a time by each drive-thru station, but only one pick-up event occurs at a time.

Conclusion:

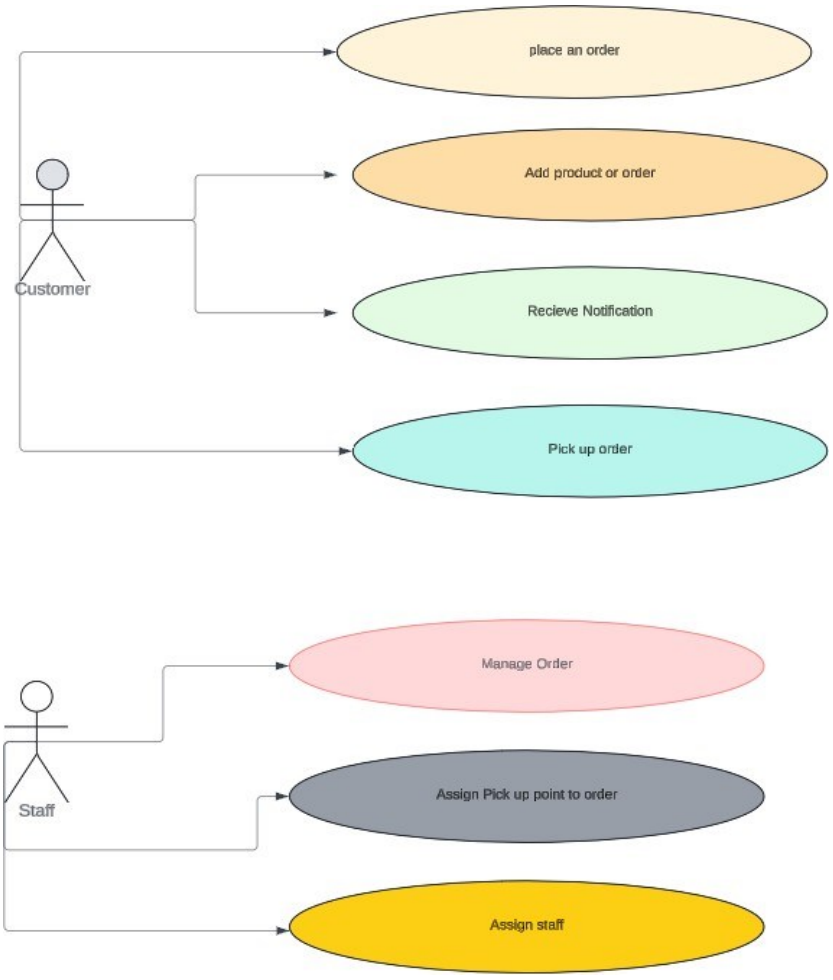
The essential elements of a drive-thru ordering system are reflected in the ERD's well-organized design.
It offers a complete foundation for a system that manages customer orders and records many facets of the ordering and pick-up procedure. The relationships specify how these entities interact, and the main entities stand in for clients, orders, merchandise, order items, and pick-up locations. The proper maintenance of the linkages between orders, customers, items, and pick-up information is made possible by this structure, which promotes effective data management and is necessary for the system to function as intended.

5.3 Sequence Diagram after Enhancement



Conclusion: The Tesco drive-thru order processing system's sequence diagram ensures efficiency, accuracy, and a better customer experience by outlining the system's clear and logical flow. It acts as a useful road map for comprehending and examining the relationships and interdependence among the components of the system.

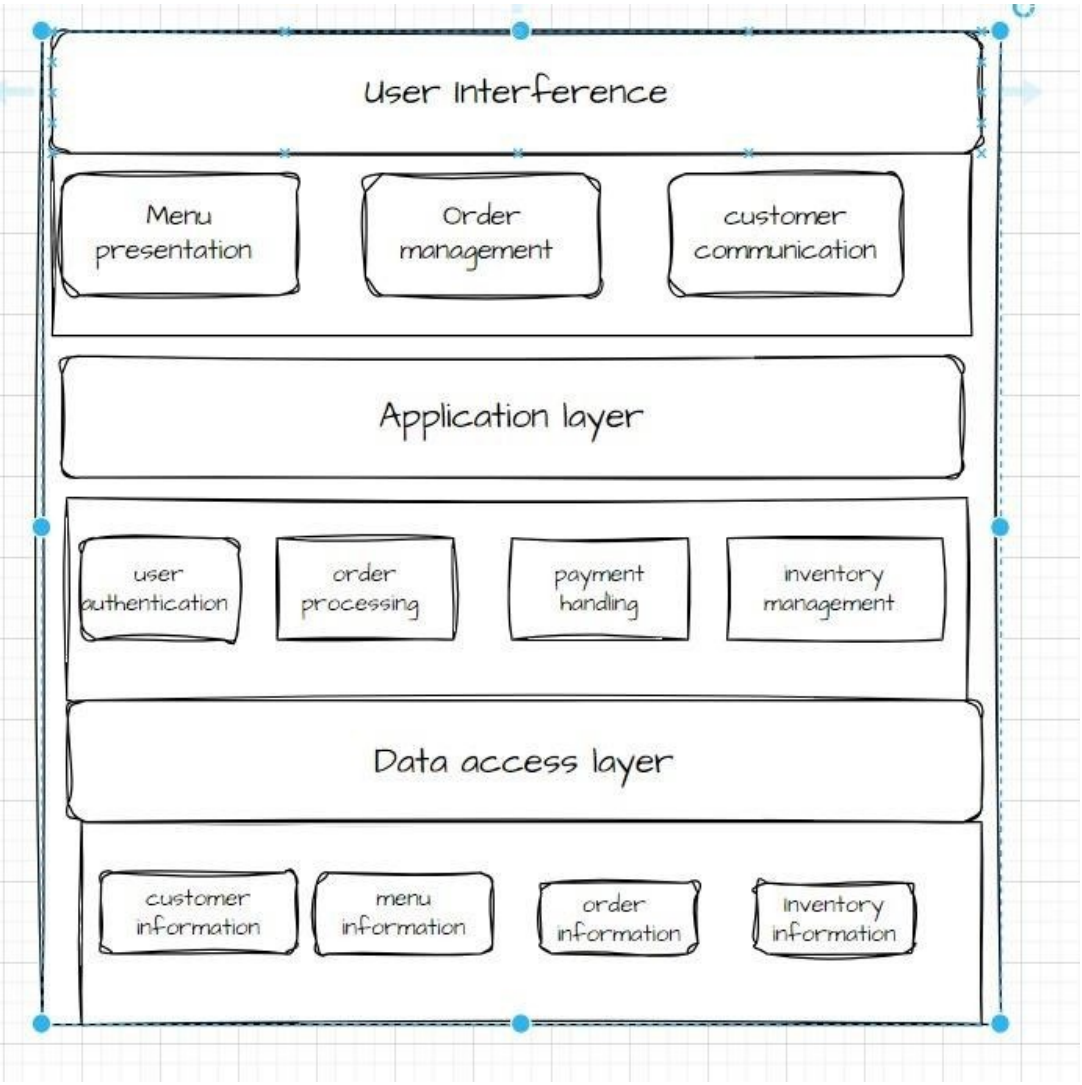
5.4 USE CASE DIAGRAM after Enhancement



Justification of Use case Diagram:

The Tesco drive-thru concept is fundamentally represented by this use case diagram, which encapsulates key features and interactions between patrons and employees. The diagram assists in the creation of a user-friendly, effective, and scalable drive-thru system that complies with stakeholder demands and company objectives by precisely outlining these interactions. This visual aid is essential for directing the design of the system, guaranteeing its proper execution, and promoting communication amongst all stakeholders

5.5 A Generic layered architecture



Architecture Layered Diagram

Architecture diagram divides the system into three separate layers: User Interference, Application Layer, and Data Access Layer. This arrangement is intended to simplify and improve the user experience.

USER INTERFERENCE:

It is essential since it deals directly with end users, which includes employees and customers, Order management, customer communication, and menu presentation are the three main elements that make up this layer. Order management makes it easier for customers to place orders, menu presentation makes it easier for customers to peruse the options available, and customer communication makes sure that there is clear and efficient communication between the customer and the service providers—all of which are crucial for a positive customer experience.

THE APPLICATION LAYER:

It serves as the cornerstone of the operational functionalities. Order processing, payment handling, inventory management, and user authentication are all included. User authentication maintains system security by verifying user identification. Order processing manages the entire life of an order creation .
Every transaction is processed swiftly and securely thanks to payment handling. Inventory control ensures that the menu items that customers see are accurate and easily accessible by keeping an eye on stock levels.

DATA ACESS LAYER:

It is in charge of storing and obtaining the vital data required for the program to function correctly. Included is data regarding orders,menu items, customers, and inventory. This layer ensures that all necessary data is accessible and updated in real-time, enabling the smooth operation of the User Interference and Application levels.

CONCULSION:

These three levels combine to offer a seamless, safe, and effective drive-thru experience, which is in line with Tesco's mission to providing prompt, dependable, and user-friendly services. This architecture supports scalability and maintainability, ensuring that the system can evolve and adapt to changing needs and technological advancements.

Structured Query Language (SQL):

```
CREATE TABLE Customer (
Customer_id INT PRIMARY KEY,
First name VARCHAR (50),
Last name VARCHAR (50),
Email VARCHAR (100),
```

```
);

CREATE TABLE Vehicle (
Vehicle_id INT PRIMARY KEY,
License_Plate VARCHAR (20),
Model VARCHAR (20),
Customer_id INT,
FOREIGN KEY (customer_id) REFERENCES Customer (Customer id)
  );

CREATE TABLE Drive_Thru (
Station_id  INT PRIMARY KEY,
Location  VARCHAR (225),
Status VARCHAR (20),
Hours_of_Operation VARCHAR (50)
);

CREATE TABLE product (
Product_id INT PRIMARY KEY,
Name VARCHAR (50),
Description TEXT
Price float
);

CREATE TABLE Inventory (
Inventory_id  INT PRIMARY KEY,
Product_id INT,
Stock_level INT,
Foreign key (product_id)  REFERENCES product (Product_id)
);

CREATE TABLE Order (
```

```
Order_id INT PRIMARY KEY,
Order_date DATE,
Total_amount FLOAT,
Customer_id INT,
FOREIGN KEY (customer_id) REFRENCES Customer (customer_id)
);

CREATE TABLE Order Item (
Order_item_id INT PRIMARY KEY,
Order_id INT,
Product_id INT,
Quantity INT,
Price FLOAT,
FOREIGN KEY (Order_id) REFRENCES order (Order_id)
FOREIGN KEY (Product_id) REFRENCES product (Product_id)
);

CREATETABLE Payment (
Payment_id INT PRIMARY KEY,
Payment_type VARCHAR (50),
Amount FLOAT,
Payment_date DATE,
Order_id INT
FOREGIN KEY (order id ) REFRENCES Order (order id ));


CREATE TABLE Order pickup (
Pickup_id INT PRIMARY KEY,
Order_id INT,
Station_id INT,
Pickup_time DATE TIME,
FOREIGN KEY (order_id) REFRENCES order (order_id),
```

```
FOREIGN KEY (station_id) REFENCES DriveThruStation (Station_id)
);

CREATE TABLE MenuItem (
Item_id INT PRIMARY KEY,
Name VARCHAR (100),
Price FLOAT
);
```

CONCULSION:

The SQL code provided describes the schema for a database intended to facilitate TESCO Drive-Thru service operations.

Records in their respective tables are uniquely identified by their Primary Keys (PK). By establishing associations between tables, foreign keys (FK) guarantee consistent and traceable references.
These connections make it easier to track orders from customers, payments, cars, inventory, and drive-thru station operations, guaranteeing a thorough and integrated system. It would be advantageous to standardize table definitions and get rid of duplicates or inconsistent information for a workable implementation.

6. References

TESCO HISTORY - Available online:

Tesco - Wikipedia

TESCO About Us- Available online:

About (tescoplc.com)

UML diagram using Lucid chart

ERD diagram using Lucid chart

Software Engineering by Ian Sommerville, publication Pearson.

Tesco Case Study Analysis (ecommerce-digest.com)