**1. Abstract:**

This is a report for Stationary Shop. It is a new startup for selling all kinds of stationary products from different suppliers. The focal point of this report is the storage view of the shop, the present situation is studied, and all the requirement specifications is produced. According to user requirements a database has been designed, highlighting all the required functions and a detailed overview of the functions that are to be implemented, and further recommendations are provided.

**2. Domain Description**

Profile and Operations

Stationary shop is a small shop run by a few people. They sell different products of stationary from last 5 months. It is a new shop with low resources.

This shop has one office:

* Management department, in Lahore.
* Logistics department, in Lahore

Manage department manage the whole shop management system. Check all the orders. It also checkout the items of stock. Take care of employees and run the system smoothly. If the items are short, they purchase more items and enlist them on website

**Current situation:**

At present condition of the shop is not so good not bad, currently there is no proper management system in the shop to store the data about the products. On the software side a stand-alone accounting package is being used in the sales office and word processing and spreadsheet facilities are available on the computer.

Logistics department is running manually and there's no automatic system in place. On receiving the inventory transport its reception is recorded on paper after which it's far saved as a result on the idea of individual experience. Due to this case shop is dealing with many troubles of loss of accuracy, gradual velocity of response and non-systematic storage allocation.

# Requirement Specification:

For the sake of clarity, the administrative department has been broken down into the following sub processes in this report:

### Sales:

### The procedure sales involve selling department of the shop dealt by the admin where orders are placed by the customers and quantity of the products within the order are updated in the store of the shop.

### Reports Generation:

### Reports related to the sales and purchase of products for a day are generated for the overall analysis of profit for the respective day.

### Purchase Products:

### An automated order for purchasing of Products is sent from the admin department to respective suppliers in case of products are out of stock.

### Reception:

### The following procedure is considered reception for this report, and it entails receiving various products from various suppliers.

### Addition of Products:

### The process involves addition of products with their initial quantity available for sale provided that the product already does not exist.

\

**Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Functions | Entities | Priority |
| *Sales* | | | |
| 1. | Insert/Update/Delete products from Orders coming from Customers. | Customers, Orders, Order-Products | High |
| 2. | Update available stock in the store department of the shop. | Orders, Order-Products, Products, Stock | High |
| 3. | Generation of total sales bill for an order placed by a customer | Order, Order-Products, Products | High |
| *Reports Generation* | | | |
| 4. | Generation of report for sales calculating the total sales for a day | Bill Reports, Orders | Normal |
| 5. | Generation of report for purchase calculating the total purchases for a day | Purchase Report, Products-Suppliers | Normal |
| 6. | View of bill and purchase report for the same day for better analysis of profit for the day | Bill report, Purchase Report | Low |
| *Purchasing Products* | | | |
| 7. | Insert/Update/Delete of Purchases made for the stock | Products, Store, Suppliers | High |
| 8. | Update available stock in the store department of the shop. | Products, Products-Suppliers, Suppliers, Stock | High |
| 9. | Generation of total purchase bill from a specific supplier. | Products, Products-Supplier, Supplier | High |
| 10. | View of all Products that are short in stock and are required to be ordered from respective suppliers. | Products, Stock | Normal |
| *Reception* | | | |
| 11. | Insert/Update/Delete of receipt of all stock available. | Supplier, Products, Stock, Supplier-Products | High |
| *Addition of Products* | | | |
| 12. | Insert/Update Products that should be available for sale | Products, Stock | High |
| 13. | List of all products available for sale. | Products | Low |

**Non-Functional Requirements:**

* Update, Deletion, Insert and View functionalities should only be available to the admin.
* Sales and Purchase Reports shall only be available to the admin and owner of the company for the better analysis of profit.
* Only authorized employees or admin should be able to use the system.
* The database must be available to Sales and Purchasing Departments.
* Reports generated should be based on days and must be accurate.

**Assumption:**

**Sales:**

* Only valid Product IDs would be considered for sales that are available in the Products.
* In case of short stock, order will not be placed for the customer until required product quantity is met.
* Total bills will be generated for all the products ordered within an order.
* Sales would be handled by the admin only.

**Report Generation:**

* Only one sales report and one purchase report would be generated for a single day.
* Reports would be generated by summing up the sales and purchases made for that respective day.
* Reports would be generated automatically once the day ends.

**Purchasing Products:**

* Out of Stock Products would already be displayed while making a purchase.
* Valid Product IDs can only be purchased. In case of new product, the product must be included in the Products table first before making a purchase.
* Total Purchase bill would be generated for all the products ordered within a day.
* Purchase and delivery of products would be handled in purchase department only.
* Purchases can be made from only the available suppliers.

**Addition of Products:**

* Only those products would be added which were not present in the database before.
* The attributes of the new product should be unique.
* The addition of products must be done by the admin only as he is the only user of the database at the present moment.

# Entity Relationship Diagram

**Diagram

Description automatically generated**

**Entities and Attributes**

**Products:**

|  |  |
| --- | --- |
| Product\_ID (primary key) | INT |
| Product\_Name | Varchar (100) |
| Unit\_price | INT |
| Available Quantity | INT |

This entity keeps the track of products where are available and their unit price and quantity.

Product\_ID assign the unique id to the products and it is selected as a primary key.

Product\_Name is for the name of the products for better understanding.

Unit\_price saves the price of the products.

Available Quantity tells about the quantity of a product which is available.

**Suppliers:**

|  |  |
| --- | --- |
| Supplier\_ID (primary key) | INT |
| Supplier\_Name | Varchar (30) |
| Supplier\_contact | INT |

Suppliers contain the Supplier\_ID which assigns the unique id to the suppliers and is set to primary key.

Supplier\_Name and contact contains the supplier details.

**Customer:**

|  |  |
| --- | --- |
| Customer\_ID (primary key) | INT |
| Customer\_Name | Varchar (30) |
| Customer\_contact | INT |

Customer contain the Customer\_ID which assigns the unique id to the Customers and is set to primary key.

Customer\_Name and contact contains the Customer details.

**Orders:**

|  |  |
| --- | --- |
| Order\_ID (Primary key) | INT |
| Customer\_ID (Foreign Key) | INT |
| Order\_Date | Varchar (20) |
| Total\_Bill | INT |

Orders contain the Order\_ID which sets the orders unique and is used as a primary key.

Customer\_ID is used as a foreign key to know about the customer against which order is placed.

Order\_Date is used to store the date on which order is placed.

Total bill contains the total bill of the order.

**Product\_Suppliers:**

|  |  |
| --- | --- |
| {Product\_ID(Foreign key) | INT |
| Supplier\_ID(Foreign key)} primary key | INT |
| SP\_price | INT |
| Quantity | INT |

IN Product\_Supplier we use both Product\_ID and Supplier\_ID as primary key.

SP\_price saves the price of the product which we taken from supplier.

Quantity saves the product quantity which is taken from supplier.

**Order\_Products:**

|  |  |
| --- | --- |
| Order\_ID (primary key) | INT |
| Customer\_ID(Foreign key ) | INT |
| Ordered\_Quantity | INT |
| Date and Time | Varchar(20) |
| Bill of product Order | INT |

**Store:**

|  |  |
| --- | --- |
| Product\_ID(Foreign Key) | INT |
| P\_quantity | INT |

This entity tracks the record of the product and its quantity.

Product\_ID is used as a foreign key.

P\_quantity stores the quantity of the product which is available at the back.

**Validation:**

As per the client prerequisites it tends to be obviously seen from the above-mentioned description of entities and their relations that the model is catching all the data that is important to the business; hence the completeness of the conceptual model can be validated against the set of user requirements.

User requirements relating to the sales are captured by the sales department. The products have a column which show its status weather it is available or not and it also show how many of the products left so only the available products shown on the database software. The user can only buy the products that are only available for sale.

Proper measurements have been taken to keep the details of products in arranged form. Further more the breakdown of the process into further sub processes has made the representation of the model very simple, hence it will be easy for the developer to implement resulting into a user-friendly system as a final product.

**Descriptions:**

Entity Relationship Diagram portraits the structure of the storage part of the proposed system. It can be spectated from the diagram that stationary storage department is performing five main functions that includes Sales, Reports Generation, Purchasing Products, Reception and Addition of Products.

After sales the reports are generated, Insertion, update and deletion of all the stock available will be handled by reception function, that will trigger the addition of products function. That process involves addition of products with their products with their initial quantity available for sale provided that the product does not exists. The reception function shall only be available to the admin. Sale and Purchase Reports shall only be available to the owner and the admin. Since it is assumed that only valid Product IDs would be considered for sales that are available in the products. And the sales would be handled by the admin only. A single report will be generated automatically for sales and one for purchases. Products that are out of stock will be displayed while making purchases. Only those products would be added which were not presented in the database before. Tracking of products is done by the product entity.

The order\_products entity is a many to many between orders and products tables in which it keeps the details of each order that is placed by customers, where the Customer\_ID is the foreign key. Date and time is also used to distinguish between the orders that are placed by each customer. Store entity keeps the record of products and their quantity. Requirements related to sales are handled by the sales department specifically.

**Screenshots of Front End:**

**Admin Registration Form**

**Graphical user interface

Description automatically generated**

**Admin Login Form**

**Graphical user interface, website

Description automatically generated**

**Add Product**

**Diagram

Description automatically generated**

**Add Employee**

**Chart

Description automatically generated with low confidence**

**Purchase Department**

**Graphical user interface

Description automatically generated with low confidence**

**Sales Department**

**Table

Description automatically generated with low confidence**

**Screenshots of Developed Reports**

**Monthly Profit Report**

**Table

Description automatically generated**

**Out of Stock Products**

**Table

Description automatically generated**

**Most Sold Product**

**Graphical user interface, table

Description automatically generated**

**Most Revenue Generating ProductGraphical user interface

Description automatically generated**

**Normalization Logical Design:**

**1st Normal Form:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Order ID** | **Order**  **Date** | **Total Sales** | **Net Profit** | **Product ID** | **Product Name** | **Sales Unit Price** | **Purchase Unit Price** | **Ordered Quantity** |
| 001 2021-12-25 | | 210 | 75 | 1001 | Papers | 70 | 45 | 3 |
| 120 | 10 | 1003 | Ink | 60 | 55 | 2 |
| 002 2021-12-25 | | 900 | 240 | 1002 | Register | 300 | 220 | 3 |
| 003 2021-12-26 | | 140 | 50 | 1001 | Papers | 70 | 45 | 2 |
| 120 | 100 | 1010 | Scale | 30 | 25 | 4 |

The following table represents the sales department (order\_product) of the Stationary Shop. As the mentioned table includes multi valued attributes hence is not in the 1st normal form.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Order ID** | **Order**  **Date** | **Total Sales** | **Net Profit** | **Product ID** | **Product Name** | **Sales Unit Price** | **Purchase Unit Price** | **Ordered Quantity** |
| 001 | 2021-12-25 | 210 | 75 | 1001 | Papers | 70 | 45 | 3 |
| 001 | 2021-12-25 | 120 | 10 | 1003 | Ink | 60 | 55 | 2 |
| 002 | 2021-12-25 | 900 | 240 | 1002 | Register | 300 | 220 | 3 |
| 003 | 2021-12-26 | 140 | 50 | 1001 | Papers | 70 | 45 | 2 |
| 003 | 2021-12-26 | 120 | 100 | 1010 | Scale | 30 | 25 | 4 |

With no multivalued attributes left, it is not in first normal form.

**2nd Normal Form:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Order ID** | **Order**  **Date** | **Total Sales** | **Net Profit** | **Product ID** | **Product Name** | **Sales Unit Price** | **Purchase Unit Price** | **Ordered Quantity** |
| 001 | 2021-12-25 | 210 | 75 | 1001 | Papers | 70 | 45 | 3 |
| 001 | 2021-12-25 | 120 | 10 | 1003 | Ink | 60 | 55 | 2 |
| 002 | 2021-12-25 | 900 | 240 | 1002 | Register | 300 | 220 | 3 |
| 003 | 2021-12-26 | 140 | 50 | 1001 | Papers | 70 | 45 | 2 |
| 003 | 2021-12-26 | 120 | 100 | 1010 | Scale | 30 | 25 | 4 |

The following table is only in first normal form but their exists some anomalies within this table.

**Update Anomaly:**

If the sales unit price or purchase unit price of Product ID 1001 is updated, it would require updating multiple records as the following Product ID was ordered in multiple orders.

**Deletion Anomaly:**

If the Product ID 1003, 1002 or 1010 are deleted from the following orders we will lose information regarding their name, sales unit price and their purchase unit price.

**Insertion Anomaly:**

If any order includes more than one product, Order ID and Order Date are repeated which causes duplication of data.

|  |  |
| --- | --- |
| **Order ID** | **Order Date** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Product ID** | **Product Name** | **Sales Unit Price** | **Purchase Unit Price** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Order ID** | **Product ID** | **Quantity Ordered** | **Total Sales** | **Net Profit** |

The following table has no partial dependencies hence is in 2nd Normal Form

**3rd Normal Form:**

There are no transitive dependencies hence the table is already in 3rd Normal Form.

**Denormalization Opportunities:**

A possible denormalization situation in the system can be in the sales department. Instead of retrieving the sales unit price and the purchase unit price of the ordered product every time from the product table and then calculating the total sales and net profit for the respective order, a better practice is to denormalize the following table.

If sales unit price and purchase unit price are made attributes of the order\_product table, calculating total sales and net profit would be quick.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order ID** | **Product ID** | **Quantity** | **Sales Unit Price** | **Total Sales** | **Net Profit** |

Denormalization

**Conclusion:**

The mentioned model relates to three different departments sales, purchase and stock department that are integrated to each other and are basic requirements of the system. With the passage of time, the presented model may require some extra specifications in the ERD related to the respective department. These changes would be directly made in the system with an update in the presented model.

**Recommendations:**

The existing system is at a very initial stage as the business is just established. The present database can fulfill the requirements of the shop until the business progresses to a larger scale. For a larger scale, the product entity can be further divided into sub entities for each product type e:g: pens, books etc. Furthermore, a security connection must be established between the admin and the system as the admin must be the only user of the database.