# **INFSCI 2710: DATABASE MANAGEMENT SYSTEM**

# **PROJECT REPORT**

# **OVER-THE-COUNTER MEDICINES**

# **PHARMACY APPLICATION MANAGEMENT SYSTEM**

**University of Pittsburgh**

**Submitted by:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SL. No.** | **Name** | **Student ID** | **Pitt Email Address** |
| 1 | Aishwarya Jakka | 4366820 | aij12@pitt.edu |
| 2 | Kwesi Randolph Aguillera | 4171544 | kra40@pitt.edu |
| 3 | Piu Mallick | 4374215 | pim16@pitt.edu |
| 4 | Shruti Gupta | 4374956 | shg104@pitt.edu |

1. **Basic Introduction**

The pharmacy application system has an inventory of Over the Counter (OTC) medicines packages, hereinafter referred simply as Medicine for brevity, categorized by disease/ailment type, that can be browsed and searched by customers on the website and can be added to the cart to make a purchase. The application has two kinds of users/actors. They are stated as follows:

1. **Users/Actors**

* **Customers** - a Person who purchase medicines.
* **Staff** - Staff can either be a salesperson or a manager.
* **Salesperson** – a Person who manage the order of customers and have access to the inventory / stock for a particular store.
* **Manager -** a Person who manage inventory and salesperson details for a particular store. They can also view the statistics (viz. sales across stores, medicine availability, top selling items(medicines) and so on.) for their stores.

The application has a login page for all types of Actors, viz. Customer and Staff. There is a role-based authentication process, which means that all the users/actors can login using the same login page, however they will be directed to a different portal after logging in, according to their access or roles.

1. **Application Features**
2. **Login/Registration**
3. All actors are able to login into the system if they have their user-id and password created.
4. After logging in they view a welcome page with features respective to their role/access.
5. The Salespersons has to create a new user account for the Customer so that the new customer can access the website.
6. **Customer Specific Features**
7. Customer’s profile will be created by Staff.
8. The customer can login into the system from the web portal.
9. After login, the customer is greeted with a ‘Welcome Message”.
10. From the welcome screen, the customer can navigate to the following pages:

* Medicines (View the list of medicines by choosing a particular store/pharmacy)
* Orders (View his/her order history)
* Cart (View or Edit Cart)
* Logout (The logout page – the customer will be able to logout)

1. He/she can check the stock availability of the medicine(s), based on store and then place order.
2. Once a Customer picks a medicine from a store and the quantity, she/he can add it to the shopping cart.
   1. The Customer can continue to shop for Medicines after adding a Medicine to the cart. The cart will continue to hold the Medicine(s) she/he has added in it.
   2. Customer can visit her/his cart anytime to check the items in the cart. The cart lists the Medicine(s) added to the cart for which the order is not yet placed.
   3. Once a medicine is chosen from a store, another store cannot be chosen. Medicine can be added to cart from one store only.
   4. In the cart the Customer cannot increase/decrease a Medicine that she/he has added before.
   5. In the cart the Customer can delete any Medicine from the cart.
3. After placement of order, he/she will receive a confirmation message.
4. **Salesperson Specific Features**
5. Salesperson’s profile will be created by Manager.
6. The salesperson can login into the system from the web portal.
7. After login, he/she can see the new orders placed by the customer. He/she can see the order history of the customers.
8. He/she can:

* View / add / update the list of medicines(products).
* View / add / update inventory details.
* View / add / update customer details.
* View/ add category of medicine
* View sale for his/her store

1. **Manager Specific Features**
2. The manager can login into the system from the web portal.
3. The manager can oversee all the details pertaining to the store.
4. The manager has all the roles which the salesperson has.
5. Additionally, he/she has a provision to see the statistical reports of the store that he/she manages.
6. **Business Rules**
   1. User can be a customer or a staff.
   2. One customer can place many orders.
   3. One order is only from one customer.
   4. One staff works in only one store.
   5. One store can have many staffs.
   6. One manager can manage only one store.
   7. One order can contain many items.
   8. One item can belong to only one order.
   9. Order\_item can contain only one medicine.
   10. One medicine can be in many order\_items.
   11. One medicine can be in many categories.
   12. One category has many medicines.
   13. One medicine can be in many stocks.
   14. One Stock has many medicines.
   15. One medicine can be from many suppliers.
   16. One supplier can have many medicines.
   17. One store can have many stocks.
   18. One stock can be in one store.
   19. One supplier can have many stocks.
   20. One stock can be from many suppliers.
   21. One order is placed to one store
   22. One store can have many orders.
   23. One manager has many salespersons.
   24. One salesperson has only one manager.
7. **Technologies Used in building the application**

* **Database:** MySQL
* **Front-end:** HTML, CSS,PHP

1. **Assumptions**
   1. The number of stores has been limited to 3.
   2. The number of suppliers has been limited to 3.
   3. No user-profile will be physically deleted from the system.
   4. For the purpose of this system, dummy data has been creation with a timeline of one year.
   5. All medicines received in stock have not expired
   6. A state tax of 7% all orders and overhead costs on inventory received is different for each supplier
   7. Manager will be added by DBA.
2. **Project Repository**

The project repository can be found in the following location:

<https://github.com/piumallick/Online-Pharmacy-Application-Management-System>

1. **ER Diagram**

See following page.

1. **Database Design**

**Files:**

* + [Create\_tables.sql](https://github.com/piumallick/Online-Pharmacy-Application-Management-System/blob/master/sql/create_tables.sql) – Table Definitions
  + [Insert.sql](https://github.com/piumallick/Online-Pharmacy-Application-Management-System/blob/master/sql/insert.sql) – Insert script to insert the dummy data
  + [Update.sql](https://github.com/piumallick/Online-Pharmacy-Application-Management-System/blob/master/sql/update.sql) – Update script added to perform calculations of the cost of inventory received adding the overhead costs and include the tax on the items sold when an order is placed.

The following tables have been created for the project:

**A screenshot of a computer

Description automatically generated**

The detailed description of each table is given below:

1. **STORE**

**TABLE STRUCTURE:**

A screenshot of a social media post

Description automatically generated

* The table ‘STORE’ contains 3 unique records.
* It contains the details of the stores, such as store id, store name and store address.
* ‘store\_id’, which is the primary key of the table STORE, is set to AUTO\_INCREMENT.
* All the fields are set to NOT NULL.

1. **CUSTOMERS**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘CUSTOMERS’ contains 13 unique records.
* It contains the details of the customers, such as cust\_id, first\_name, last\_name, email\_address, passwd, phone\_number, address, gender and dob (date of birth).
* Gender ‘F’ refers to female customers, whereas gender ‘M’ refers to male customers.
* ‘cust\_id’, which is the primary key of the table CUSTOMERS, is set to AUTO\_INCREMENT.
* All the fields are set to NOT NULL.
* ‘email\_address’ and ‘phone\_number’ have unique constraints.

1. **STAFF**

**TABLE STRUCTURE:**

A screenshot of a social media post

Description automatically generated

* The table ‘STAFF’ contains 9 unique records.
* Staff can be either a manager or a salesperson: Manager is identified by the role ‘M’ and Salesperson is identified by the role ‘S’.
* It contains the details of the staff, such as staff\_id, first\_name, last\_name, email\_address, passwd, phone\_number, ssn, date\_of\_joining, salary, store\_id and role.
* ‘staff\_id’, which is the primary key of the table STAFF, is set to AUTO\_INCREMENT.
* All the fields are set to NOT NULL.
* ‘email\_address’ and ‘phone\_number’ have unique constraints.

1. **MEDICINE**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘MEDICINE’ contains 110 unique records.
* It contains the details of all the OTC (Over the Counter) medicines, such as medicine\_id, medicine\_name and medicine\_desc (medicine description).
* ‘medicine\_id’, which is the primary key of the table MEDICINE, is set to AUTO\_INCREMENT.
* All the fields are set to NOT NULL.

1. **CATEGORY**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘CATEGORY’ contains 10 unique records.
* It contains the details of all the medicine categories, such as category\_id, category\_name, lower\_age, upper\_age and gender.
* ‘category\_id’, which is the primary key of the table CATEGORY, is set to AUTO\_INCREMENT.
* All the fields are set to NOT NULL.

1. **MEDICINE\_CATEGORY**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘MEDICINE\_CATEGORY’ contains 121 unique records.
* It contains the details of all the medicines and its categories. The details are stored in the fields like medicine\_id and category\_id.
* ‘medicine\_id’ and ‘category\_id’ are both defined as the primary keys of the table MEDICINE\_CATEGORY, and hence they can be called composite primary keys.
* All the fields are set to NOT NULL.

1. **SUPPLIER**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘SUPPLIER’ contains 3 unique records.
* It contains the details of all the suppliers who supply medicines to different stores. It contains different attributes like supplier\_id, supplier\_name, address, phone\_number and email\_address.
* ‘‘supplier\_id’, which is the primary key of the table SUPPLIER, is set to AUTO\_INCREMENT.
* All the fields are set to NOT NULL.
* ‘email\_address’ and ‘phone\_number’ have unique constraints.

1. **STOCK**

**TABLE STRUCTURE**

A screenshot of a cell phone

Description automatically generated

* The table ‘STOCK’ contains 18 unique records.
* The table is being used to store the bulk orders to a particular supplier. A stock can consist of many medicines ordered in bulk from the same supplier.
* It contains different attributes like stock\_id, supply\_date, overhead\_pct, total\_cost and supplier\_id.
* ‘‘stock\_id’, which is the primary key of the table STOCK, is set to AUTO\_INCREMENT.
* All the fields are set to NOT NULL.
* The attribute ‘total\_cost’ is set to a default value of 0.

1. **HAS\_STOCK\_SUPPLY**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘HAS\_STOCK\_SUPPLY’ contains 330 unique records.
* This table contains the list of medicines supplied by “a supplier” and the quantity purchased and the cost price. This table would contain and would be useful when there is a need to know “How much of the medicine has been sold?”
* It contains different attributes like stock\_id, supplier\_id, unit\_cost\_price, medicine\_id, manufacture\_date, expiry\_date, quantity and total\_cost.
* ‘stock\_id’, ‘supplier\_id’ and ‘medicine\_id’ are all defined as the primary keys of the table HAS\_STOCK\_SUPPLY, and hence they can be called composite primary keys.
* All the fields are set to NOT NULL.
* The columns ‘unit\_cost\_price’, ‘quantity’ and ‘total\_cost’’ have been set to a default value of 0.

1. **HAS\_STORE\_STOCK**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘HAS\_STORE\_STOCK’ contains 330 unique records.
* This table is used to divide the stock among 3 stores.
* It contains different attributes like stock\_id, stock\_id, medicine\_id, availability\_of\_medicine and unit\_selling\_price.
* ‘store\_id’, ‘stock\_id’ and ‘medicine\_id’ are all defined as the primary keys of the table HAS\_STORE\_STOCK, and hence they can be called composite primary keys.
* All the fields are set to NOT NULL.
* The column ‘unit\_selling\_price’ was set to a default value of 0.

1. **ORDERS**

**TABLE STRUCTURE:**

A screenshot of a cell phone

Description automatically generated

* The table ‘ORDERS’ contains 66 unique records.
* This table is used to track the order details of each customer pertaining to various stores.
* It contains different attributes like order\_id, total\_amt, order\_date, cust\_id and store\_id.
* ‘order\_id’, which is the primary key of the table ORDERS, is set to AUTO\_INCREMENT
* ‘cust\_id’ and ‘store\_id’ are the Foreign Key Constraints which refer to the CUSTOMERS and STORE tables respectively.
* All the fields are set to NOT NULL.

1. **ORDER\_ITEMS**

**TABLE STRUCTURE**

A screenshot of a cell phone

Description automatically generated

* The table ‘ORDER\_ITEMS’ contains 330 unique records.
* This table is used to track the list of the medicines in an order.
* It contains different attributes like order\_id, medicine\_id, stock\_id, unit\_selling\_price, quantity and total\_amt.
* ‘order\_id’, ‘medicine\_id’ and ‘stock\_id’ are all defined as the primary keys of the table. ORDER\_ITEMS, and hence they can be called composite primary keys.
* All the fields are set to NOT NULL.
* The fields ‘unit\_selling\_price’, ‘quantity’ and ‘total\_amt’ are set to default value of 0 at the first place.

After doing the initial insertion of data into the tables (**insert.sql**), following update scripts(**update.sql**) were executed. The following code is also available in the Github repository (link mentioned previously in the report).

/\* UPDATE ON HAS\_STOCK\_SUPPLY TOTAL\_COST WHERE TOTAL\_COST = UNIT\_COST\_PRICE \* QUANTITY\*/

UPDATE HAS\_STOCK\_SUPPLY SET total\_cost= (quantity\*unit\_cost\_price);

/\* UPDATE ON STOCK TOTAL\_COST WHERE TOTAL\_COST = SUM(TOTAL\_AMT) \* (100\*OVERHEAD\_PCT/100) \*/

UPDATE STOCK

SET total\_cost = (SELECT SUM(HAS\_STOCK\_SUPPLY.total\_cost)

FROM HAS\_STOCK\_SUPPLY

WHERE FROM HAS\_STOCK\_SUPPLY

WHERE HAS\_STOCK\_SUPPLY.supplier\_id = STOCK.supplier\_id

GROUP BY HAS\_STOCK\_SUPPLY.stock\_id);

UPDATE STOCK

SET total\_cost = total\_cost \* (1 + overhead\_pct\*0.01);

/\* UPDATE ON HAS\_STORE\_STOCK UNIT\_SELLING PRICE = HAS\_STOCK\_SUPPLY.UNIT\_COST\_PRICE \* 1.15 \*/

UPDATE HAS\_STORE\_STOCK

SET UNIT\_SELLING\_PRICE = ((SELECT UNIT\_COST\_PRICE

FROM HAS\_STOCK\_SUPPLY

WHERE HAS\_STOCK\_SUPPLY.stock\_id = HAS\_STORE\_STOCK.stock\_id

and HAS\_STOCK\_SUPPLY.medicine\_id = HAS\_STORE\_STOCK.medicine\_id) \* 1.15);

/\* UPDATE ON ORDER\_ITEMS TOTAL\_AMT WHERE TOTAL\_AMT= UNIT\_SELLING PRICE\_PRICE \* QUANTITY\*/

UPDATE ORDER\_ITEMS SET total\_amt = quantity \* unit\_selling\_price;

/\* UPDATE ON ORDERS TOTAL\_AMT WHERE TOTAL\_AMT= (SUM(total\_amt)) \* 1.07 where we add \*/

UPDATE ORDERS

SET total\_amt= (SELECT SUM(total\_amt)

FROM ORDER\_ITEMS WHERE ORDER\_ITEMS.order\_id = ORDERS.order\_id) \* 1.07;

1. **Application Screenshots & Flow**
   1. **User Log in Page**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Customer Welcome Page**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Available Medicines for Customers**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Customer Order History**

**A screenshot of a cell phone

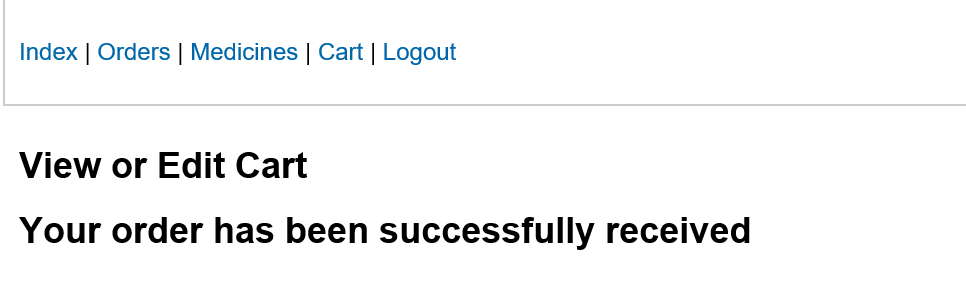
Description automatically generated**

* 1. **Customer Shopping Cart**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Customer Submit order**



* 1. **Staff Welcome Page (for Salesperson)**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Staff Welcome Page (for Store Manager)**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Customer Creation Page**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Staff Creation Page for a Manager**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Edit Customer Details**

**A screenshot of a social media post

Description automatically generated**

* 1. **Edit Staff Details for the Manager**

**A screenshot of a cell phone

Description automatically generated**

* 1. **List Customer Details**

**A screenshot of a cell phone

Description automatically generated**

* 1. **List Store Staff**

**A screenshot of a cell phone

Description automatically generated**

* 1. **List Stocks**

**A screenshot of a cell phone

Description automatically generated**

* 1. **List Medicines**

**A screenshot of a cell phone

Description automatically generated**

* 1. **List Suppliers**

**A screenshot of a cell phone

Description automatically generated**

* 1. **List Store Stock**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Edit Medicine**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Edit Supplier**

**A screenshot of a social media post

Description automatically generated**

* 1. **Add Medicine Information**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Add Stock Information**

**A screenshot of a cell phone

Description automatically generated**

* 1. **Add Supplier Information**

**A screenshot of a social media post

Description automatically generated**

* 1. **Store Sales Order History**

**A screenshot of a social media post

Description automatically generated**

* 1. **Statistical Reports for Store Manager**
* **Report Dashboard**

**A screenshot of a social media post

Description automatically generated**

* **Store Sales Comparison**

**A screenshot of a cell phone

Description automatically generated**

* **Most Purchased Items by Customers**

**A screenshot of a cell phone

Description automatically generated**

* **Medicine backlog in the store**

**A screenshot of a social media post

Description automatically generated**

* **Low Availability Medicines in the store**

**A screenshot of a cell phone

Description automatically generated**

* **Availability of Medicines in each store**
* **CVS Pharmacy**

**A screenshot of a cell phone

Description automatically generated**

* **Giant Eagle Pharmacy**

**A screenshot of a cell phone

Description automatically generated**

* **Rite Aid Pharmacy**

**A screenshot of a cell phone

Description automatically generated**

1. **Testing Efforts**

* Unit Testing and Peer Testing have been done across various pages of the application. Certain bugs and merge conflicts were experienced at various steps of the coding and testing phase, which have been resolved in the testing phase.

1. **Limitations**
   * 5 medicines can be added to a stock at one time.
   * Adding a medicine, category, customer and staff can only be done one at a time
   * Editing of medicine, category, stock, customer and staff can only be done one at a time
2. **Future Improvements** 
   * Allow searchable statistics for manager can be implemented.
   * Have custom search feature for customer and staff to search for a stock, order or a medicine.
   * The old data for customers and staffs can be moved to archival tables instead of deleting.
   * The application can be improvised in future by implementing recommendation features using Data Mining algorithms and Machine Learning techniques on a real-time dataset.