**Functional Requirements**

***Overview:***

The functional requirements of our system describe several different types of users, each with a distinct set of actions available to them. As each different user will have different uses for the system, they will be presented with a different set of options. Having set up functionality specific to each type of user also allows us to keep data hidden from users who don’t have full permission. This allows our single database to be seen through multiple different “views”, each meeting the requirements of the user in question.

***Notes on the menu system:***

The menu system will first prompt the user to choose which type of user they are. Once their type has been chosen the user will be prompted with a list of different types of queries they can make. They choose by typing in a number from 1 to n, where n is the number of query choices. If the user types 0, they will be taken to the previous menu. If the chosen query needs any additional information such as the value of a variable, then the user will be prompted to enter this. The relevant SQL statement will then be executed by the system. In this way, our system exposes queries to the user through entered numbers, along with relevant required information inputted by the user.

***Users:***

We have three different types of employees: cashiers, pharmacists and administrators. We also have one set of requirements representing the options for all customers. The following is a description of each user’s requirements and the expected behaviour of the system:

***Cashiers*** deal with a small window of our company. They spend all of their time at the register and because of this they only need to be able to do a few things on a day to day basis. Their most common tasks are to get the price of a product and record the sale of a product. When the cashier checks the price, they will need to check if the product is a prescription. Assuming no prior database knowledge, we have made it easy for the cashier to select which query they want based on a simple menu system of options. The system will execute the appropriate prewritten SQL statement based on which option was chosen by the cashier. In the case of the cashier getting the price based on product ID, they will enter the ID after choosing this menu option. This ID will be read as a variable and will be passed to the prewritten SQL statement.

***Pharmacists*** need to view a queue of prescriptions to be filled. They are able to cancel prescriptions and add prescriptions. Also, when a prescription is picked up they will select the menu option for removing from it the prescription queue. The pharmacist will then enter the ID of which prescription to remove and the prewritten SQL query will take this variable and execute the statement.

***Administrators*** need to manage and view the pharmacy’s sales data in several different ways. The administrator will be able to see all of the products that have ever been sold, as well as all products that share a price with any other product to allow them to compare product values. The administrator will also be able to see the products with the greatest price. To display products of specific stock values, the admin can enter these stock values in that will execute a query to output these products. Another type of query would be to view all products in a certain department that have a stock level in a certain range (which the admin will input). To keep track of employee sales, the admin can also see all products that have ever been sold by a specified employee. While testing out his complex query skills, the administrator also supports a query that shows all customers who have prescriptions for every prescription product in the database. Finally, the administrator can manually edit stock levels of products in case a discrepancy is detected after a physical inventory check.

***Customers*** can see a wide variety of product information as well as information pertaining to their specific prescription orders. For products they can see a view ordered by a variety of attributes. These include by price, categories, tags and stock level. After they choose this menu option they have a secondary option to choose which attribute to sort by. The customer can then see the output of this query. A customer wanting to see the lowest prices will find this functionality invaluable when making shopping decisions.