PHARMACY MANAGEMENT

Submitted to: Dr. Sahar Fawzi

Eng. Ahmed Elbaz

Eng. Asmaa Mohamed

Submitted by: Yomna Osama Hussein

Group 7

ID: 1190203

Email: yomna.osamma@gmail.com

Submitted in: 31/5/2020

1 TABLE OF CONTENTS

List of figures:

2 INTRODUCTION:

Let me start with the contents of the project, which includes 3 tables. The first table provides two "Product Identification and Pharmaceutical Price" columns. The second table includes three columns "Customer ID, total number of previous orders, and total number of medications in all previous orders." The last table includes three columns, Patient ID, Product ID, and Health Care Coverage Rebate (percent), on which one person will have several rows for the various medications he is buying.

This drug administration is combined with a variety of alternatives. So we can load pharmacy data into two ways, whether manually or by loading it from an excel file. The first three choices are adding a row in each table. The fourth alternative, whether we add a row first or not, we can load data from an excel sheet, any data entered is saved, we can even save a table in a new excel sheet after the user takes the file name. After taking from the user about the appropriate table to show, we can display the data in this table.

We can also compute new tables. The ordered drugs table which includes 4 columns (Customer ID, Drug ID, Drug price, and Drug price after discount). The second table that contains 3 columns (Drug ID, total number of ordered drugs, and total price). The last table which computes the extra discount, which is explained in the following table, contains 5 columns (The new extra discount table contains Customer ID, Total amount of all previous orders, Total amount of all current orders, extra discount, and Total amount of all current orders after the extra discount). The user can also choose

whether he wants to display the tables on the command window or saving it to an excel file, after providing its name.

Table 1 Extra Discount

Price (LE)	Extra Discount
>1000	25%
>500	20%
>300	15%
>150	10%
>100	5%

3 USER MANUAL

The pharmacy management is converted to a graphical user interface 'GUI' as seen in the figure below, making it easier for the user to use.
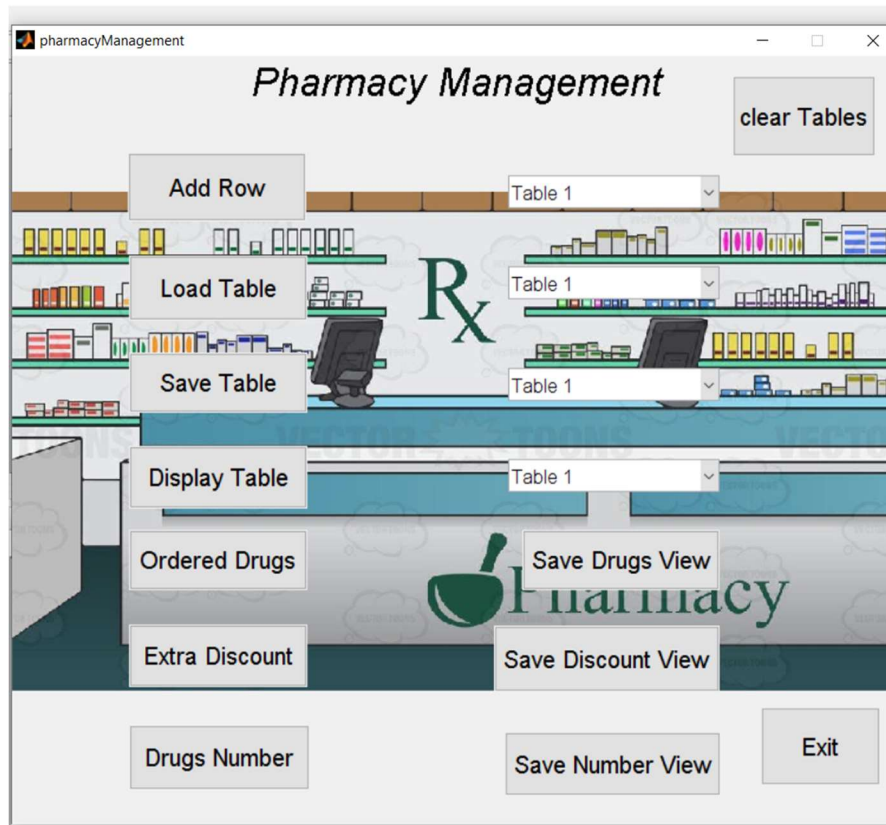

Figure 3-1 Pharmacy Management GUI

3.1 ADD ROW BUTTON:

By pressing on the first button 'Add Row' a new GUI is called to add a new column in the first table, choosing table 1 by default, we can change the table chosen from the pop-up menu. In the first table, when the user enters the drug ID the GUI code will check first if it is repeated or not. If it is repeated, the data beside this ID will be updated with the new one added and the table will be sorted. If the user entered a

negative value in any text box, a message box will appear to reject this input. If the user chooses to add a row in table 2, the same steps will be repeated. If the user chooses to add a new row in table 3, no negative values is accepted and the discount range is between 0% and 100% and the figures below shows these steps.
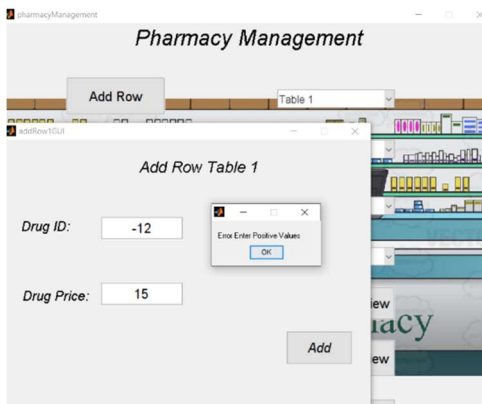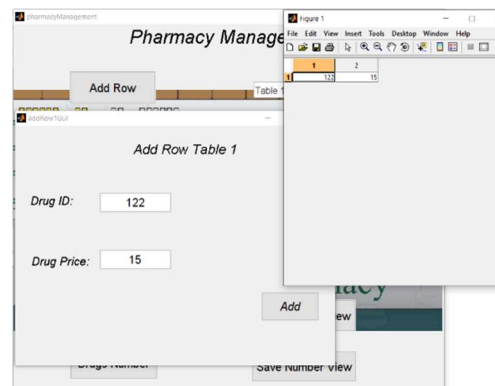

Figure 3-2 Negative ID Table 1
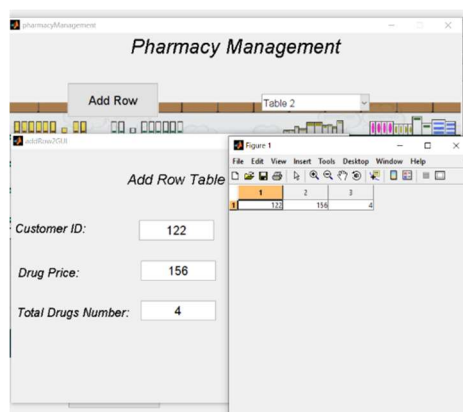

Figure 3-3 Correct input Table 1
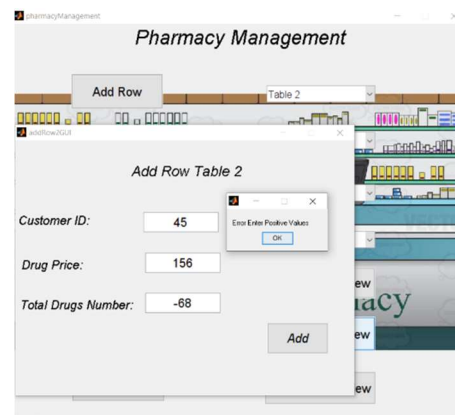

Figure 3-4 Correct Input Table 2
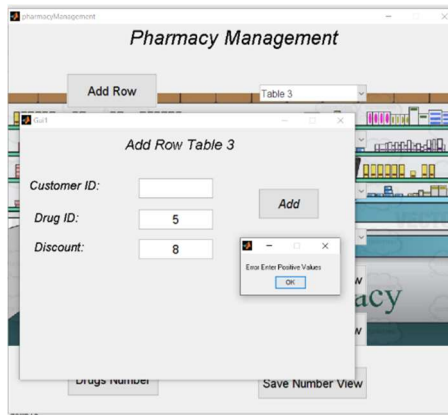

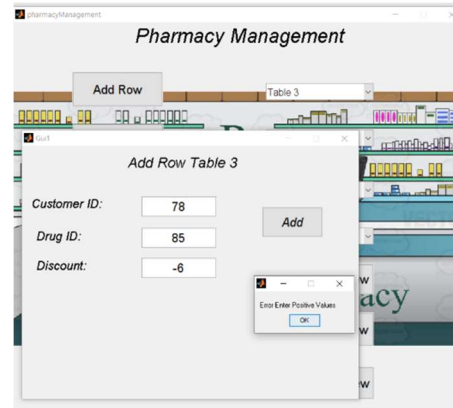Figure 3-5 Negative Input Table 2

Figure 3-6 Empty Text Box Table 3

Figure 3-7 Wrong Discount Range

3.2 LOAD TABLE BUTTON:

By pressing on the second button 'Load Table' a browse will appear to get the path of the file and load data on table 1 by default. If we choose another table from the pop-up menu, the same sequence will be repeated. If the user didn't choose any path, an error message will appear to choose a correct path. If the use chooses to enter to table 1 an excel sheet containing more than two columns or less than them, an error message will appear. If the user chooses a correct path of an excel matching the dimensions of the table, the table will appear sorted. The same sequence is applied when the user chooses table 2 or table 3. The figures below show all of the sequences of the Load button.
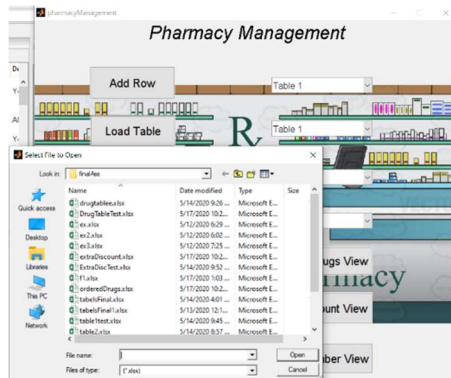

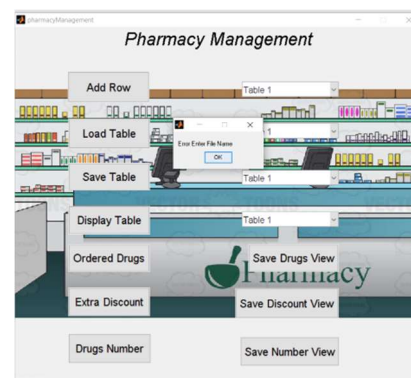Figure 3-8 Opening Browse after pressing load
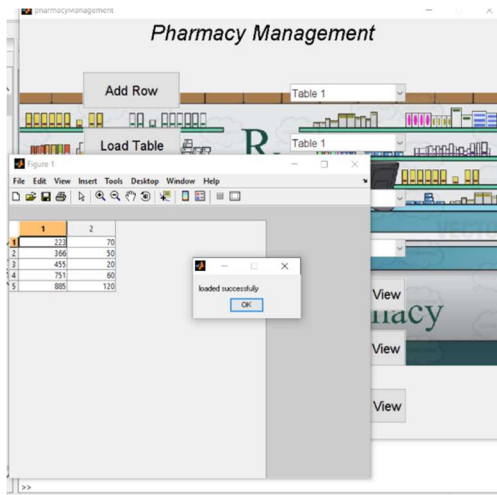

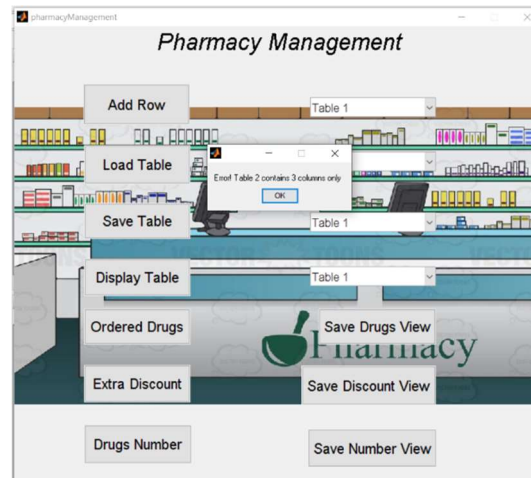Figure 3-9 No chosen file name

Figure 3-10 Loaded Successfully

Figure 3-11 Incorrect Dimensions

3.3 SAVE TABLE BUTTON:

By pressing on the save button, a browse will appear to choose a file to save in. If no data in the table chosen, a message box will appear containing 'No Data to be saved'. The table chosen can be changed from the pop-up menu. If there is no file name inserted, an error message will appear. The following figures shows all of this steps.
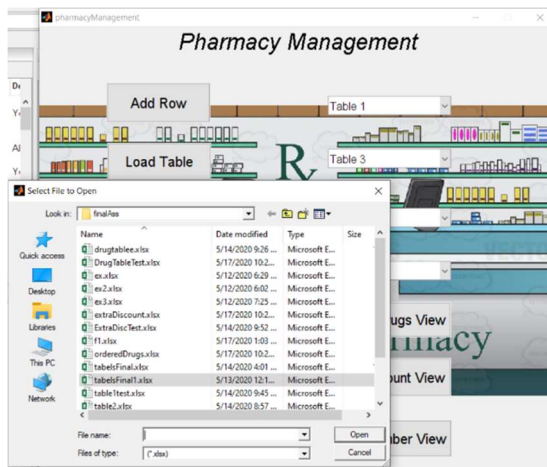

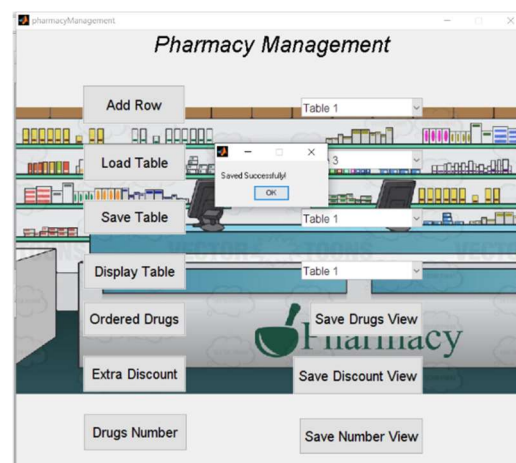Figure 3-12 Pressing Save Button
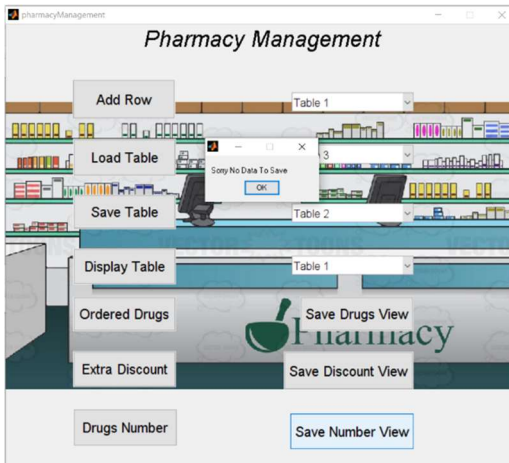

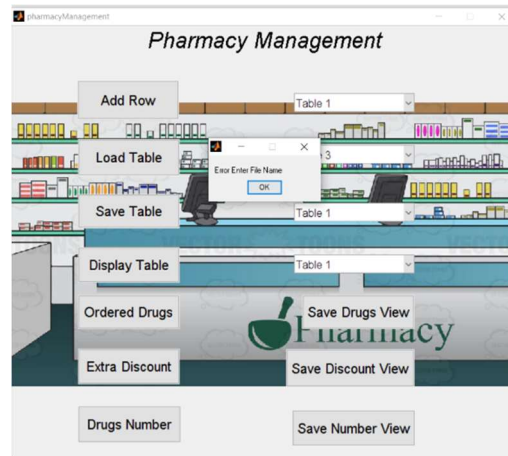Figure 3-13 Saved Successfully

Figure 3-14 Table is empty

Figure 3-15 Empty file name

3.4 DISPLAY TABLE BUTTON:

After pressing on this button, Table 1 chosen by default will appear. After choosing the table number from the pop-up menu, the table chosen will appear. If the table chosen is an empty table, error message will appear as shown in the figures below.
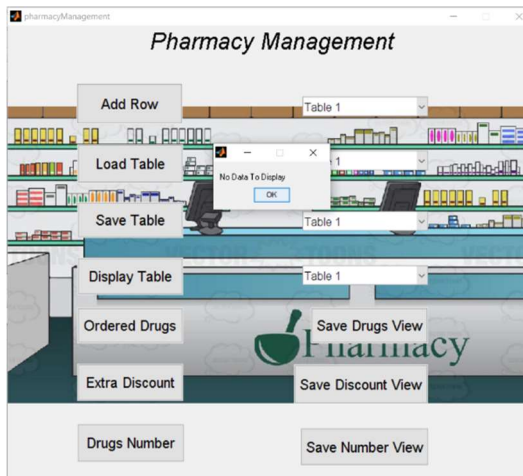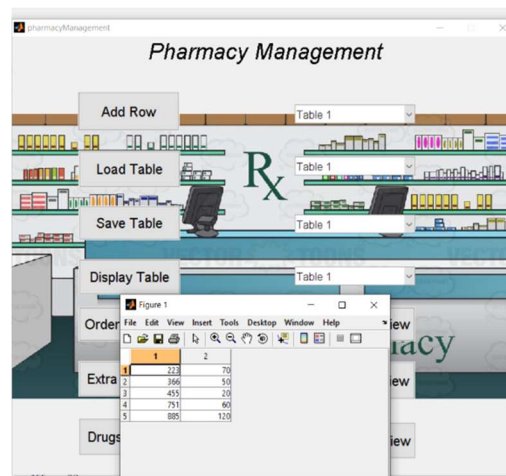

Figure 3-16 Display an Empty Table

Figure 3-17 Displayed Successfully

3.5 ORDERED DRUGS BUTTON:

Ordered drugs table is computed using table 1 and table 3 contains 4 columns as mentioned before. If the user pressed the button and no data whether in table 1 or table 3, error message will appear. If the user pressed the button and there is a data in the two tables, the table will appear. If the user wants to save the view, he can press 'Save Drugs View' button and a browse will appear to take the path name as shown in the figures below. If the file name is empty, an error message will appear as mentioned before. If the user chose the 'Save Drugs View' button first the table is computed successfully and saved with the same sequence mentioned in this paragraph.
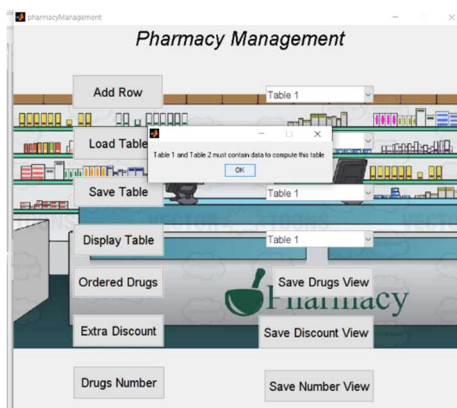

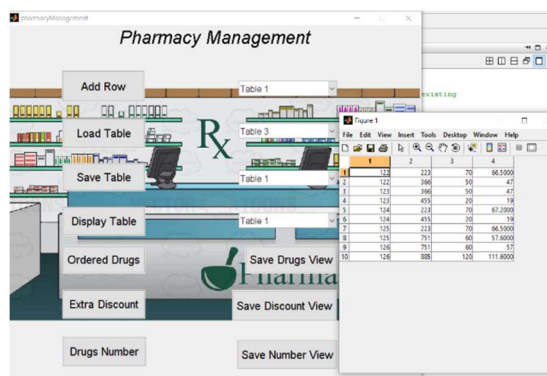*Figure 3-18 No data in tables 1 and 2*

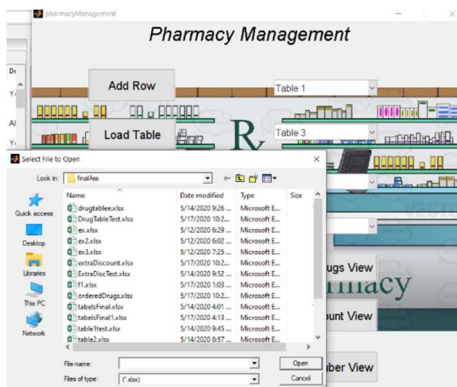*Figure 3-19 Computed Successfully*

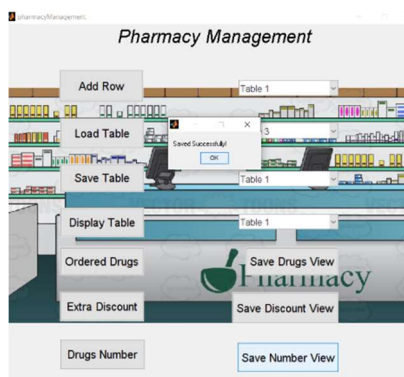*Figure 3-20 Press Save Drugs Button*

Figure 3-21 Save Drugs View

3.6 EXTRA DISCOUNT BUTTON:

This table needs the previous table and table 2 to be computed. Table 1 and table 3 are needed to compute the previous table. So this table needs the data inside the three tables. So if the user pressed this button and tables are empty, error message will appear. If the user pressed the button and there is a data in the three tables, the computed table will appear. If the user wants to save the view, he can press 'Save Discount View' button and a browse will appear to take the path name as shown in the figures below. If the file name is empty, an error message will appear as mentioned before. If the user chose the 'Save Discount View' button first the table is computed successfully and saved with the same sequence mentioned in this paragraph.
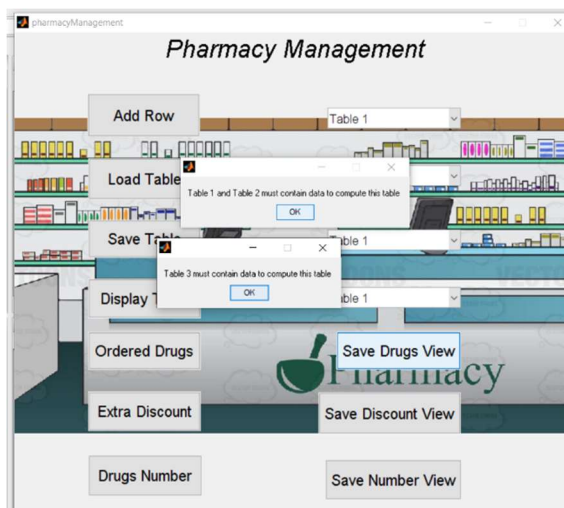

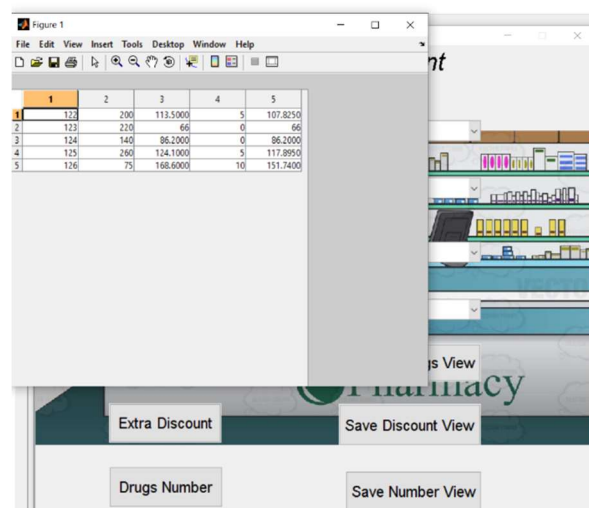*Figure 3-22 No Data in the tables*

Figure 3-23 Extra discount Table

3.7 DRUGS NUMBER BUTTON:

Drugs Number table is computed using table 1 and table 3 contains 3 columns as mentioned before. If the user pressed the button and no data whether in table 1 or table 3, error message will appear. If the user pressed the button and there is a data in the two tables, the table will appear. If the user wants to save the view, he can press 'Save Number View' button and a browse will appear to take the path name as shown in the figures below. If the file name is empty, an error message will appear as mentioned before. If the user chose the 'Save Number View' button first the table is computed successfully and saved with the same sequence mentioned in this paragraph.
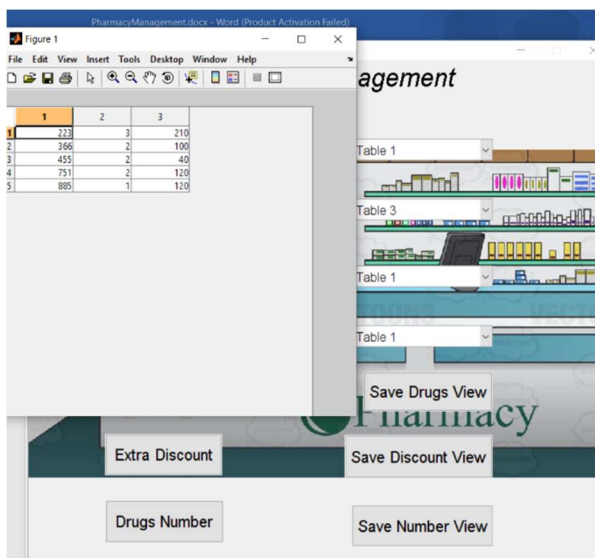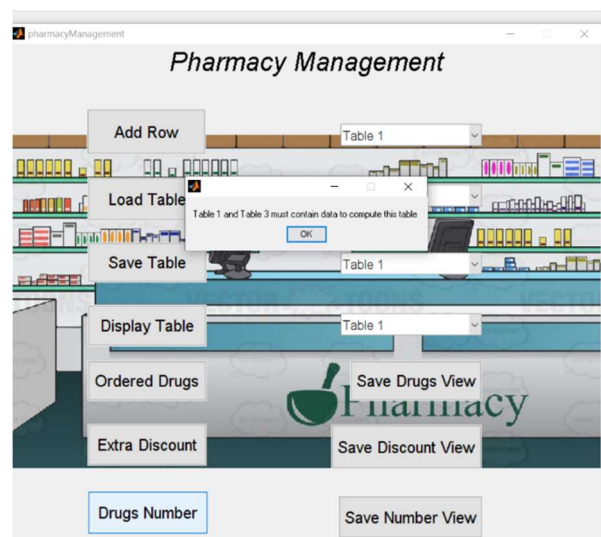

Figure 3-24 Drugs Number Table

Figure 3-25 No Data inside the tables

We can clear tables by pressing on "Clear Tables" button. Finally, user can exit the program after pressing the "Exit" button.

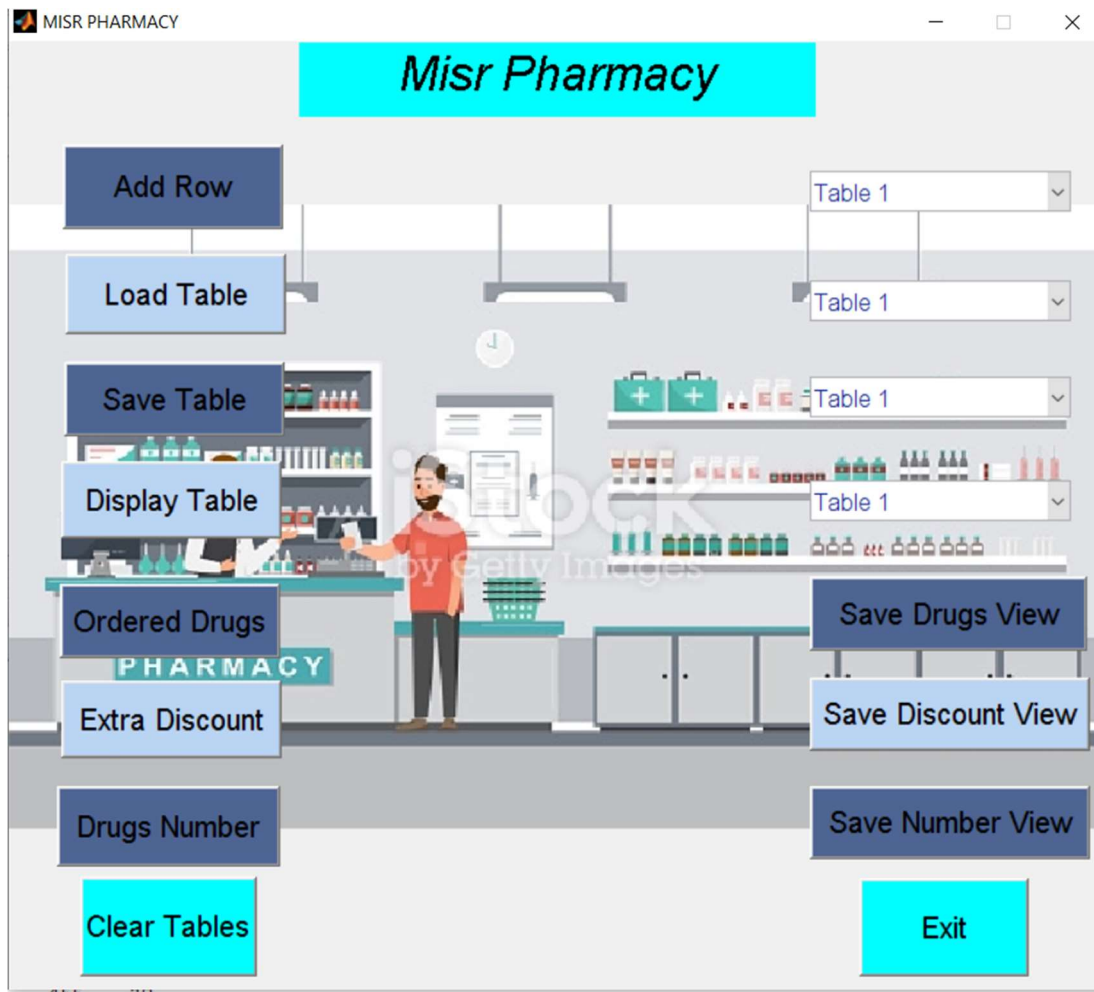Then I changed the background of the program for better look as shown in the figure below.
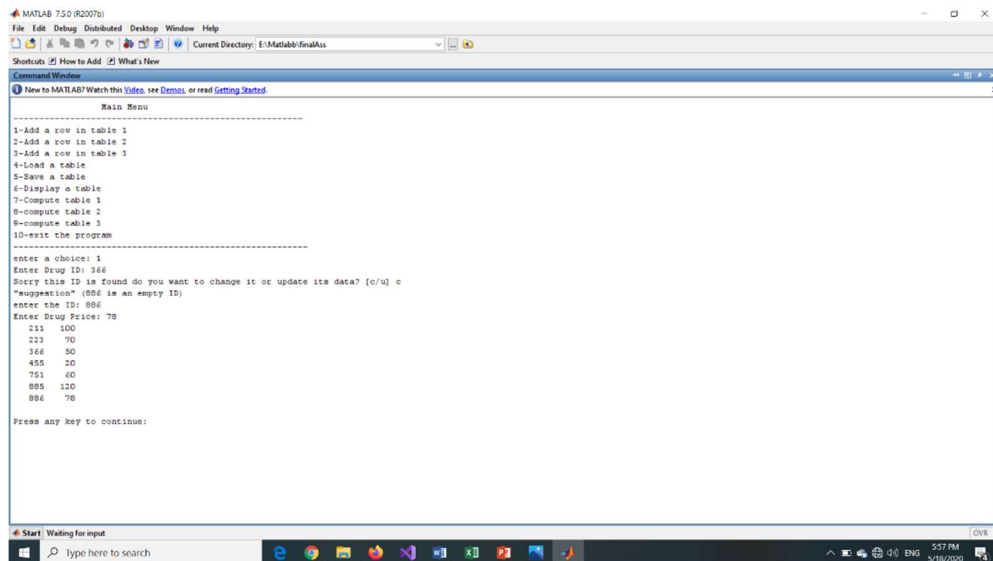

Figure 3-26 Final GUI

4 CODE IMPLEMENTATION:

The Pharmacy Administration system offers multiple options. In the first three tables the first choice is to add a new row. I used several functions to lower errors to add a new row. Drug and consumer Identities can't be replicated in the first and third columns. Therefore, if the user has entered a repeated ID, he will be asked whether he wants to modify the entered ID or update this ID info. If he wants to modify the ID entered, a suggestion for an empty ID would appear as seen in the figures below.

Figure 4-1 repeated ID

When I changed the code to a GUI, I used new functions as 'global' to make the variables used as table 1 and table 2 accessed among .m files. 'uitable(f,'Data',table,'Position',[0,0,400,400])' to display a table in a GUI. I used the same code functions to check for a recurring id, and whether to modify it or update this id's info.

in the load, save, and display buttons the same functions are used except the input function

which is changed in GUI to str2double(get('handles.tagname','string')). At which set takes the

input from the user in GUI dynamic text box as string and str2double change it to double. To

display an output in a GUI I used set(handles.tagname,value). Load, Save and display I used the

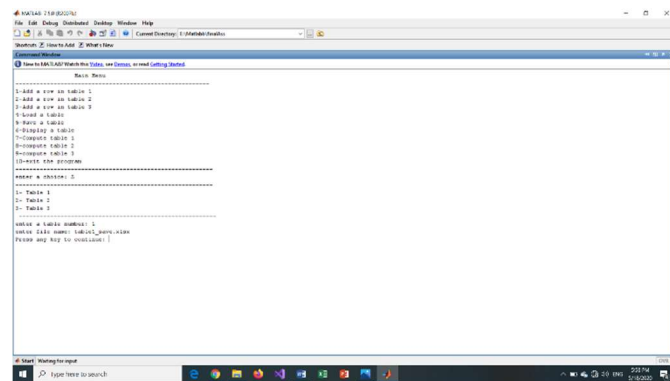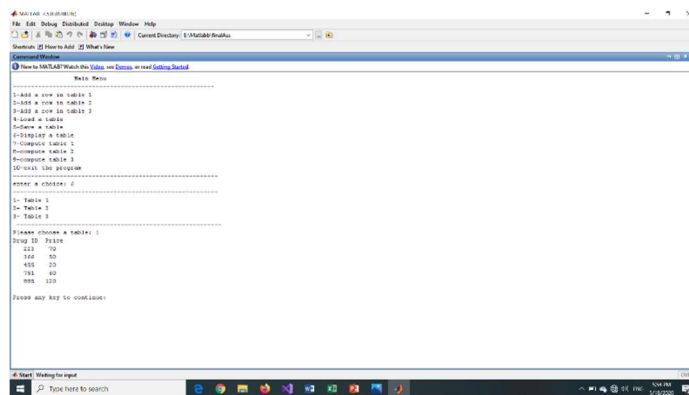same functions and algorithm as shown in the figures below.

Figure 4-2 Save Function

Figure 4-3 Display

Figure 4-4 Extra Discount table