IT0011 – Integrative Programming and Technologies

Final Project Specifications

1. Front Page
2. Introduction
3. Objectives
4. Screen Shots of Story Board
5. Summary

Program Specification

1. Design a good layout
2. Provide Good Images (if applicable)
3. With Data Persistence (MySQL)

Submit a soft copy of the following:

1. Documentation (pdf)
2. Python Source Code (Project Files)



Department of Information Technology

IT0011: INTEGRATIVE PROGRAMMING AND TECHNOLOGIES

SECTION

Final Project

Grade

Submitted by:



Abrero, Ezekiel





Saltarin, Ashley Argay

Poniado, Mark Joseph

Guillermo, Justine Rome

|  |
| --- |
| Submitted to:  Dr. Deanghel, Roman. |

**DOCUMENT CONTENTS:**

1. **INTRODUCTION**

Our group created an inventory drug system. The system will help people, especially in the medical field. The old fashion way of counting the inventory is labor-intensive and costly. Humans are often prone to error, and overcounting or undercounting objects. Each time an item is added into the inventory drug system, several information items are recorded in the system. The expiration dates are one of those. They are conveniently accessible to know what items will come upon expiration. Keeping all the details of the drug items in one place also helps if a product is ever recalled. You will know exactly how much of that drug you have on hand by simply clicking the mouse.

Having a drug inventory system in place to monitor your drug items can also help in your investment if you are owning a drug store. Confusion and theft are less likely to happen when you know exactly what products are in storage and what products have been ordered and delivered. The drug inventory system is created with Python and PHP. To store the data of the items we connected python to the PHP database with the use of it we successfully store the item id, item name, item brand, item price, stock amount and item expiration. Having this inventory drug system that we created there will be fewer unnecessary items in storage and allow for a more organized item.

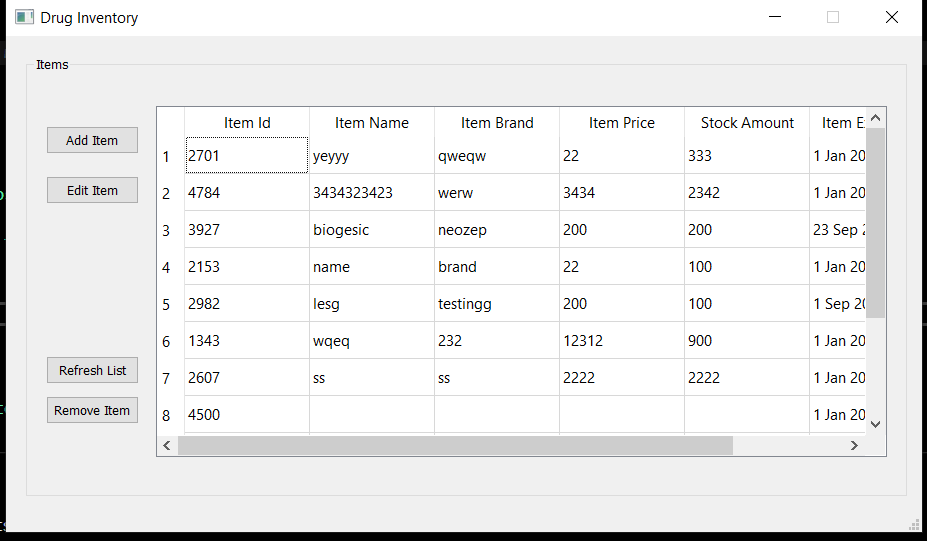
1. **OBJECTIVES**

The system objective is to help and make the life of people in the medicine field or a drug store owner much easier. This will help those people in the medical fields and for those who owns a drug store to ensure that the need of the customers will be met and to also to track if the drugs in the inventory is still good for ordering or delivery. Our drug inventory system main objectives are the following:

* Help organize various type of drugs
* Providing a function that will let the user add, remove or edit an item
* Tracking of orders of drugs
* To ensure continuous supply of drugs
* To avoid both overstocking and under-stocking of inventory
* To eliminate duplication in ordering or replenishing stocks
* To design proper organization for inventory
* Facilitates data furnishing for short-term and long-term inventory preparation and monitoring.

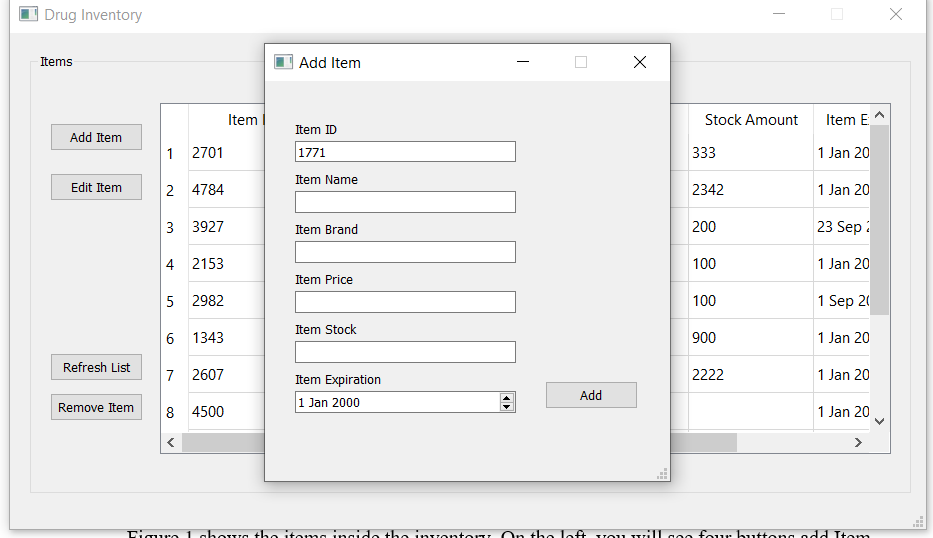
1. **PROGRAM SCREEN SHOTS**

Capture the program window for each process that has been taken. Write some description about the captured screen.



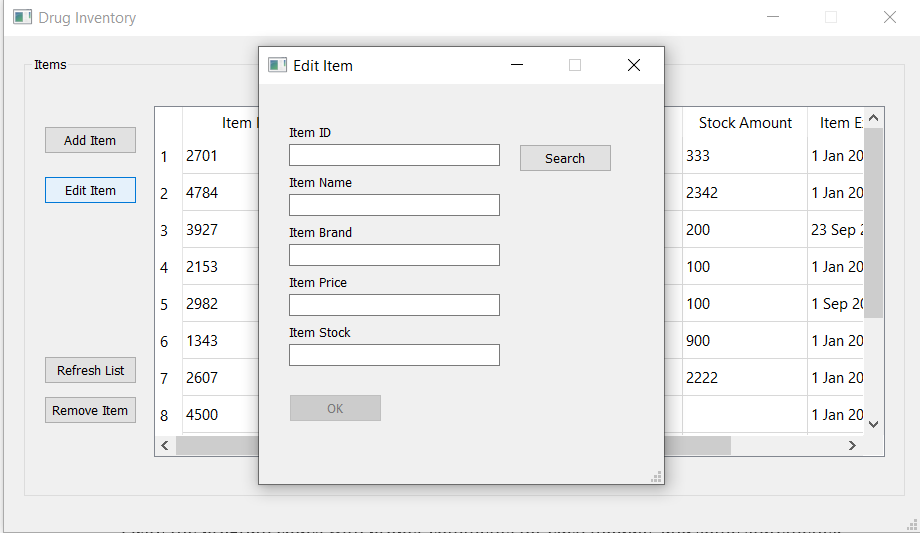
**Figure 1. Main Program**

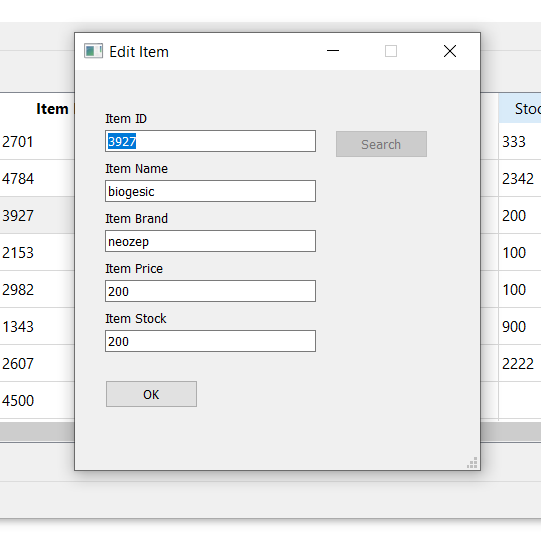
Figure 1 shows the items inside the inventory. On the left you will see four buttons add Item, edit Item, refresh list and remove item.



**Figure 2. Add Item**

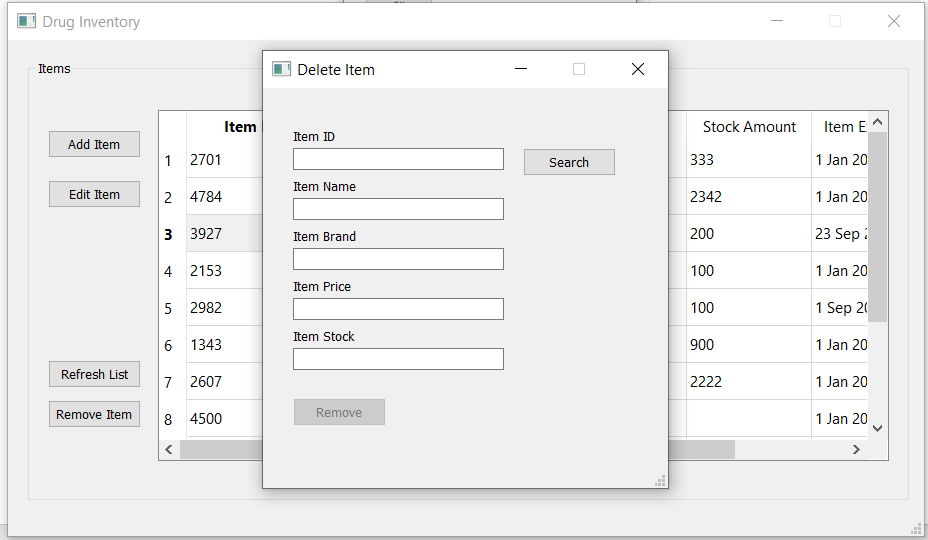
Figure 2 shows up when you press the add item button, in this window the item ID is automatically generated to avoid item\_Id duplication.





**Figure 3. Edit Item**

Figure 3 shows up when you press the edit item button, in this window you will type the item id for the search and if the id exist the item will be shown and you can edit the item name, rand, price, and stock.



**Figure 4. Remove item**

Figure 4 shows up when you press the remove item button, in this window you will type the item id for the search and if the id exist the item will be shown and you can delete the item that you have searched.

1. **SOURCE CODE**

*from* PyQt5 *import* QtCore, QtGui, QtWidgets

*import* sys

*import* pymysql

*import* random

*import* ast

*#below this is the ui*

class Ui\_editItem(*object*): *#editItem UI*

**def** searchClicked(*self*):

        print('search')

        q **=** "SELECT \* FROM inventory"

**try**:

            cur.execute(q)

            result **=** cur.fetchall()

**for** row **in** result:

**if** row[0] **==** *self*.idText.text():

*self*.search.setEnabled(False)

*self*.idText.setReadOnly(True)

*self*.okButton.setEnabled(True)

*self*.itemText.setText(row[1])

*self*.brandText.setText(row[2])

*self*.priceText.setText(row[3])

*self*.stockText.setText(row[4])

            print('wtf')

**except**:

            print('no such record')

**def** okClicked(*self*):

        print('ok')

        item **=** *self*.itemText.text()

        brand **=** *self*.brandText.text()

        price **=** *self*.priceText.text()

        stock **=** *self*.stockText.text()

        q **=** "UPDATE inventory SET item\_name=%s,item\_brand=%s,item\_price=%s,stock\_amount=%s WHERE item\_id = %s"

**try**:

            print("execute")

            value **=** (item, brand, price, stock, *self*.idText.text())

            cur.execute(q, value)

            print("commit")

            db.commit()

            print('added to db')

**except** Exception:

            db.rollback()

*self*.centralwidget.close()

**def** setupUi(*self*, **editItem**):

        editItem.setObjectName("editItem")

        editItem.resize(405, 400)

        editItem.setMinimumSize(QtCore.QSize(405, 400))

        editItem.setMaximumSize(QtCore.QSize(405, 400))

*self*.centralwidget **=** QtWidgets.QWidget(editItem)

*self*.centralwidget.setObjectName("centralwidget")

*self*.label **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label.setGeometry(QtCore.QRect(30, 40, 55, 16))

*self*.label.setObjectName("label")

*self*.label\_4 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_4.setGeometry(QtCore.QRect(30, 190, 81, 16))

*self*.label\_4.setObjectName("label\_4")

*self*.label\_2 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_2.setGeometry(QtCore.QRect(30, 90, 81, 16))

*self*.label\_2.setObjectName("label\_2")

*self*.label\_3 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_3.setGeometry(QtCore.QRect(30, 140, 81, 16))

*self*.label\_3.setObjectName("label\_3")

*self*.label\_5 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_5.setGeometry(QtCore.QRect(30, 240, 81, 16))

*self*.label\_5.setObjectName("label\_5")

*self*.okButton **=** QtWidgets.QPushButton(*self*.centralwidget)

*self*.okButton.setGeometry(QtCore.QRect(30, 310, 93, 28))

*self*.okButton.setObjectName("okButton")

*self*.okButton.setEnabled(False)

*self*.okButton.clicked.connect(*self*.okClicked)

*self*.search **=** QtWidgets.QPushButton(*self*.centralwidget)

*self*.search.setGeometry(QtCore.QRect(260, 60, 93, 28))

*self*.search.setObjectName("search")

*self*.search.clicked.connect(*self*.searchClicked)

*self*.idText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.idText.setGeometry(QtCore.QRect(30, 60, 211, 22))

*self*.idText.setObjectName("idText")

*self*.itemText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.itemText.setGeometry(QtCore.QRect(30, 110, 211, 22))

*self*.itemText.setObjectName("itemText")

*self*.brandText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.brandText.setGeometry(QtCore.QRect(30, 160, 211, 22))

*self*.brandText.setObjectName("brandText")

*self*.priceText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.priceText.setGeometry(QtCore.QRect(30, 210, 211, 22))

*self*.priceText.setObjectName("priceText")

*self*.stockText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.stockText.setGeometry(QtCore.QRect(30, 260, 211, 22))

*self*.stockText.setObjectName("stockText")

        editItem.setCentralWidget(*self*.centralwidget)

*self*.menubar **=** QtWidgets.QMenuBar(editItem)

*self*.menubar.setGeometry(QtCore.QRect(0, 0, 405, 26))

*self*.menubar.setObjectName("menubar")

        editItem.setMenuBar(*self*.menubar)

*self*.statusbar **=** QtWidgets.QStatusBar(editItem)

*self*.statusbar.setObjectName("statusbar")

        editItem.setStatusBar(*self*.statusbar)

*self*.retranslateUi(editItem)

        QtCore.QMetaObject.connectSlotsByName(editItem)

**def** retranslateUi(*self*, **editItem**):

        \_translate **=** QtCore.QCoreApplication.translate

        editItem.setWindowTitle(\_translate("editItem", "Edit Item"))

*self*.label.setText(\_translate("editItem", "Item ID"))

*self*.label\_4.setText(\_translate("editItem", "Item Price"))

*self*.label\_2.setText(\_translate("editItem", "Item Name"))

*self*.label\_3.setText(\_translate("editItem", "Item Brand"))

*self*.label\_5.setText(\_translate("editItem", "Item Stock"))

*self*.okButton.setText(\_translate("editItem", "OK"))

*self*.search.setText(\_translate("editItem", "Search"))

class Ui\_deleteItem(*object*): *#deleteitItem UI ##----- delete querry work on it*

**def** searchClicked(*self*):

        print('search')

        q **=** "SELECT \* FROM inventory"

**try**:

            cur.execute(q)

            result **=** cur.fetchall()

**for** row **in** result:

**if** row[0] **==** *self*.idText.text():

*self*.search.setEnabled(False)

*self*.idText.setReadOnly(True)

*self*.okButton.setEnabled(True)

*self*.itemText.setText(row[1])

*self*.brandText.setText(row[2])

*self*.priceText.setText(row[3])

*self*.stockText.setText(row[4])

**except**:

            print('no such record')

**def** okClicked(*self*):

        print('remove')

        id **=** *self*.idText.text()

        q **=** "DELETE FROM inventory WHERE item\_id = %s"

**try**:

            print("execute")

            cur.execute(q, id)

            print("commit")

            db.commit()

            print('deleted from inventory')

**except** Exception **as** e:

            db.rollback()

            print(e)

*self*.centralwidget.close()

**def** setupUi(*self*, **deleteItem**):

        deleteItem.setObjectName("deleteItem")

        deleteItem.resize(405, 400)

        deleteItem.setMinimumSize(QtCore.QSize(405, 400))

        deleteItem.setMaximumSize(QtCore.QSize(405, 400))

*self*.centralwidget **=** QtWidgets.QWidget(deleteItem)

*self*.centralwidget.setObjectName("centralwidget")

*self*.label **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label.setGeometry(QtCore.QRect(30, 40, 55, 16))

*self*.label.setObjectName("label")

*self*.label\_4 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_4.setGeometry(QtCore.QRect(30, 190, 81, 16))

*self*.label\_4.setObjectName("label\_4")

*self*.label\_2 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_2.setGeometry(QtCore.QRect(30, 90, 81, 16))

*self*.label\_2.setObjectName("label\_2")

*self*.label\_3 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_3.setGeometry(QtCore.QRect(30, 140, 81, 16))

*self*.label\_3.setObjectName("label\_3")

*self*.label\_5 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_5.setGeometry(QtCore.QRect(30, 240, 81, 16))

*self*.label\_5.setObjectName("label\_5")

*self*.okButton **=** QtWidgets.QPushButton(*self*.centralwidget)

*self*.okButton.setGeometry(QtCore.QRect(30, 310, 93, 28))

*self*.okButton.setObjectName("okButton")

*self*.okButton.setEnabled(False)

*self*.okButton.clicked.connect(*self*.okClicked)

*self*.search **=** QtWidgets.QPushButton(*self*.centralwidget)

*self*.search.setGeometry(QtCore.QRect(260, 60, 93, 28))

*self*.search.setObjectName("search")

*self*.search.clicked.connect(*self*.searchClicked)

*self*.idText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.idText.setGeometry(QtCore.QRect(30, 60, 211, 22))

*self*.idText.setObjectName("idText")

*self*.itemText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.itemText.setGeometry(QtCore.QRect(30, 110, 211, 22))

*self*.itemText.setObjectName("itemText")

*self*.brandText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.brandText.setGeometry(QtCore.QRect(30, 160, 211, 22))

*self*.brandText.setObjectName("brandText")

*self*.priceText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.priceText.setGeometry(QtCore.QRect(30, 210, 211, 22))

*self*.priceText.setObjectName("priceText")

*self*.stockText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.stockText.setGeometry(QtCore.QRect(30, 260, 211, 22))

*self*.stockText.setObjectName("stockText")

        deleteItem.setCentralWidget(*self*.centralwidget)

*self*.menubar **=** QtWidgets.QMenuBar(deleteItem)

*self*.menubar.setGeometry(QtCore.QRect(0, 0, 405, 26))

*self*.menubar.setObjectName("menubar")

        deleteItem.setMenuBar(*self*.menubar)

*self*.statusbar **=** QtWidgets.QStatusBar(deleteItem)

*self*.statusbar.setObjectName("statusbar")

        deleteItem.setStatusBar(*self*.statusbar)

*self*.retranslateUi(deleteItem)

        QtCore.QMetaObject.connectSlotsByName(deleteItem)

**def** retranslateUi(*self*, **deleteItem**):

        \_translate **=** QtCore.QCoreApplication.translate

        deleteItem.setWindowTitle(\_translate("deleteItem", "Delete Item"))

*self*.label.setText(\_translate("deleteItem", "Item ID"))

*self*.label\_4.setText(\_translate("deleteItem", "Item Price"))

*self*.label\_2.setText(\_translate("deleteItem", "Item Name"))

*self*.label\_3.setText(\_translate("deleteItem", "Item Brand"))

*self*.label\_5.setText(\_translate("deleteItem", "Item Stock"))

*self*.okButton.setText(\_translate("deleteItem", "Remove"))

*self*.search.setText(\_translate("deleteItem", "Search"))

class Ui\_addItem(*object*): *#addItem UI*

**def** okClicked(*self*):

        print("ok")

        id **=** *self*.idText.text()

        item **=** *self*.nameText.text()

        brand **=** *self*.brandText.text()

        price **=** *self*.priceText.text()

        stock **=** *self*.stockText.text()

        expiration **=** *self*.dateEdit.text()

        q **=** "INSERT INTO inventory (`item\_id`, `item\_name`, `item\_brand`, `item\_price`, `stock\_amount`, `item\_expiration`) VALUES ('{}', '{}', '{}', '{}', '{}', '{}') ".format(id, item, brand, price, stock, expiration )

**try**:

            print("execute")

            cur.execute(q)

            print("commit")

            db.commit()

            print('added to db')

**except** Exception:

            db.rollback()

            print()

**def** setupUi(*self*, **addItem**):

        addItem.setObjectName("addItem")

        addItem.resize(405, 400)

        addItem.setMinimumSize(QtCore.QSize(405, 400))

        addItem.setMaximumSize(QtCore.QSize(405, 400))

*self*.centralwidget **=** QtWidgets.QWidget(addItem)

*self*.centralwidget.setObjectName("centralwidget")

*self*.label **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label.setGeometry(QtCore.QRect(30, 40, 55, 16))

*self*.label.setObjectName("label")

*self*.idText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.idText.setGeometry(QtCore.QRect(30, 60, 221, 21))

*self*.idText.setReadOnly(True)

*self*.idText.setObjectName("idText")

*self*.idText.setText(str(random.randint(1000,5000)))

*self*.nameText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.nameText.setGeometry(QtCore.QRect(30, 110, 221, 22))

*self*.nameText.setObjectName("nameText")

*self*.brandText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.brandText.setGeometry(QtCore.QRect(30, 160, 221, 22))

*self*.brandText.setObjectName("brandText")

*self*.label\_4 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_4.setGeometry(QtCore.QRect(30, 190, 81, 16))

*self*.label\_4.setObjectName("label\_4")

*self*.priceText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.priceText.setGeometry(QtCore.QRect(30, 210, 221, 22))

*self*.priceText.setObjectName("priceText")

*self*.label\_2 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_2.setGeometry(QtCore.QRect(30, 90, 81, 16))

*self*.label\_2.setObjectName("label\_2")

*self*.label\_3 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_3.setGeometry(QtCore.QRect(30, 140, 81, 16))

*self*.label\_3.setObjectName("label\_3")

*self*.stockText **=** QtWidgets.QLineEdit(*self*.centralwidget)

*self*.stockText.setGeometry(QtCore.QRect(30, 260, 221, 22))

*self*.stockText.setObjectName("stockText")

*self*.label\_5 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_5.setGeometry(QtCore.QRect(30, 240, 81, 16))

*self*.label\_5.setObjectName("label\_5")

*self*.label\_6 **=** QtWidgets.QLabel(*self*.centralwidget)

*self*.label\_6.setGeometry(QtCore.QRect(30, 290, 91, 16))

*self*.label\_6.setObjectName("label\_6")

*self*.dateEdit **=** QtWidgets.QDateEdit(*self*.centralwidget)

*self*.dateEdit.setGeometry(QtCore.QRect(30, 310, 221, 22))

*self*.dateEdit.setObjectName("dateEdit")

*self*.okButton **=** QtWidgets.QPushButton(*self*.centralwidget)

*self*.okButton.setGeometry(QtCore.QRect(280, 300, 93, 28))

*self*.okButton.setObjectName("okButton")

*self*.okButton.clicked.connect(*self*.okClicked)

        addItem.setCentralWidget(*self*.centralwidget)

*self*.menubar **=** QtWidgets.QMenuBar(addItem)

*self*.menubar.setGeometry(QtCore.QRect(0, 0, 405, 26))

*self*.menubar.setObjectName("menubar")

        addItem.setMenuBar(*self*.menubar)

*self*.statusbar **=** QtWidgets.QStatusBar(addItem)

*self*.statusbar.setObjectName("statusbar")

        addItem.setStatusBar(*self*.statusbar)

*self*.retranslateUi(addItem)

        QtCore.QMetaObject.connectSlotsByName(addItem)

**def** retranslateUi(*self*, **addItem**):

        \_translate **=** QtCore.QCoreApplication.translate

        addItem.setWindowTitle(\_translate("addItem", "Add Item"))

*self*.label.setText(\_translate("addItem", "Item ID"))

*self*.label\_2.setText(\_translate("addItem", "Item Name"))

*self*.label\_3.setText(\_translate("addItem", "Item Brand"))

*self*.label\_4.setText(\_translate("addItem", "Item Price"))

*self*.label\_5.setText(\_translate("addItem", "Item Stock"))

*self*.label\_6.setText(\_translate("addItem", "Item Expiration"))

*self*.okButton.setText(\_translate("addItem", "Add"))

class Ui\_MainWindow(*object*): *#main window*

**def** MyConverter(*self*, **mydata**):

**def** cvt(**data**):

**try** :

**return** ast.literal\_eval(data)

**except**:

**return** str(data)

**return** tuple(map(cvt, mydata))

**def** refreshData(*self*):

*self*.itemTable.setRowCount(0)

        q **=** "select \* from inventory"

        cur.execute(q)

        data **=** cur.fetchall()

**for** row **in** data:

*self*.addTable(*self*.MyConverter(row))

**def** addTable(*self*, **columns**):

        rowPosition **=** *self*.itemTable.rowCount()

*self*.itemTable.insertRow(rowPosition)

**for** i, columns **in** enumerate(columns):

*self*.itemTable.setItem(rowPosition, i, QtWidgets.QTableWidgetItem(str(columns)))

**def** addItemClicked(*self*):

*self*.window **=** QtWidgets.QMainWindow()

*self*.ui **=** Ui\_addItem()

*self*.ui.setupUi(*self*.window)

*self*.window.show()

**def** editItemClicked(*self*):

*self*.window **=** QtWidgets.QMainWindow()

*self*.ui **=** Ui\_editItem()

*self*.ui.setupUi(*self*.window)

*self*.window.show()

**def** removeItemClicked(*self*):

        print("deleteitem")

*self*.window **=** QtWidgets.QMainWindow()

*self*.ui **=** Ui\_deleteItem()

*self*.ui.setupUi(*self*.window)

*self*.window.show()

**def** refreshClicked(*self*):

        print("refresh")

*self*.refreshData()

**def** setupUi(*self*, **MainWindow**):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(916, 496)

        MainWindow.setMinimumSize(QtCore.QSize(916, 496))

        MainWindow.setMaximumSize(QtCore.QSize(916, 496))

*self*.centralwidget **=** QtWidgets.QWidget(MainWindow)

*self*.centralwidget.setObjectName("centralwidget")

*self*.viewItems **=** QtWidgets.QGroupBox(*self*.centralwidget)

*self*.viewItems.setGeometry(QtCore.QRect(20, 20, 881, 441))

*self*.viewItems.setObjectName("viewItems")

*self*.itemTable **=** QtWidgets.QTableWidget(*self*.viewItems)

*self*.itemTable.setGeometry(QtCore.QRect(130, 50, 731, 351))

*self*.itemTable.setColumnCount(6)

*self*.itemTable.setObjectName("itemTable")

*self*.itemTable.setRowCount(0)

        item **=** QtWidgets.QTableWidgetItem()

*self*.itemTable.setHorizontalHeaderItem(0, item)

        item **=** QtWidgets.QTableWidgetItem()

*self*.itemTable.setHorizontalHeaderItem(1, item)

        item **=** QtWidgets.QTableWidgetItem()

*self*.itemTable.setHorizontalHeaderItem(2, item)

        item **=** QtWidgets.QTableWidgetItem()

*self*.itemTable.setHorizontalHeaderItem(3, item)

        item **=** QtWidgets.QTableWidgetItem()

*self*.itemTable.setHorizontalHeaderItem(4, item)

        item **=** QtWidgets.QTableWidgetItem()

*self*.itemTable.setHorizontalHeaderItem(5, item)

*self*.addItem **=** QtWidgets.QPushButton(*self*.viewItems)

*self*.addItem.setGeometry(QtCore.QRect(20, 70, 93, 28))

*self*.addItem.setObjectName("addItem")

*self*.addItem.clicked.connect(*self*.addItemClicked)

*self*.editItem **=** QtWidgets.QPushButton(*self*.viewItems)

*self*.editItem.setGeometry(QtCore.QRect(20, 120, 93, 28))

*self*.editItem.setObjectName("editItem")

*self*.editItem.clicked.connect(*self*.editItemClicked)

*self*.removeItem **=** QtWidgets.QPushButton(*self*.viewItems)

*self*.removeItem.setGeometry(QtCore.QRect(20, 340, 93, 28))

*self*.removeItem.setObjectName("removeItem")

*self*.removeItem.clicked.connect(*self*.removeItemClicked)

*self*.refresh **=** QtWidgets.QPushButton(*self*.viewItems)

*self*.refresh.setGeometry(QtCore.QRect(20, 300, 93, 28))

*self*.refresh.setObjectName("refresh")

*self*.refresh.clicked.connect(*self*.refreshClicked)

        MainWindow.setCentralWidget(*self*.centralwidget)

*self*.menubar **=** QtWidgets.QMenuBar(MainWindow)

*self*.menubar.setGeometry(QtCore.QRect(0, 0, 916, 26))

*self*.menubar.setObjectName("menubar")

        MainWindow.setMenuBar(*self*.menubar)

*self*.statusbar **=** QtWidgets.QStatusBar(MainWindow)

*self*.statusbar.setObjectName("statusbar")

        MainWindow.setStatusBar(*self*.statusbar)

*self*.retranslateUi(MainWindow)

        QtCore.QMetaObject.connectSlotsByName(MainWindow)

**def** retranslateUi(*self*, **MainWindow**):

        \_translate **=** QtCore.QCoreApplication.translate

        MainWindow.setWindowTitle(\_translate("MainWindow", "Drug Inventory"))

*self*.viewItems.setTitle(\_translate("MainWindow", "Items"))

        item **=** *self*.itemTable.horizontalHeaderItem(0)

        item.setText(\_translate("MainWindow", "Item Id"))

        item **=** *self*.itemTable.horizontalHeaderItem(1)

        item.setText(\_translate("MainWindow", "Item Name"))

        item **=** *self*.itemTable.horizontalHeaderItem(2)

        item.setText(\_translate("MainWindow", "Item Brand"))

        item **=** *self*.itemTable.horizontalHeaderItem(3)

        item.setText(\_translate("MainWindow", "Item Price"))

        item **=** *self*.itemTable.horizontalHeaderItem(4)

        item.setText(\_translate("MainWindow", "Stock Amount"))

        item **=** *self*.itemTable.horizontalHeaderItem(5)

        item.setText(\_translate("MainWindow", "Item Expiration"))

*self*.addItem.setText(\_translate("MainWindow", "Add Item"))

*self*.editItem.setText(\_translate("MainWindow", "Edit Item"))

*self*.removeItem.setText(\_translate("MainWindow", "Remove Item"))

*self*.refresh.setText(\_translate("MainWindow", "Refresh List"))

*self*.refreshData()

**def** openMainWindow():*#opens the main window*

    app **=** QtWidgets.QApplication(sys.argv)

    MainWindow **=** QtWidgets.QMainWindow()

    ui **=** Ui\_MainWindow()

    ui.setupUi(MainWindow)

    MainWindow.show()

    sys.exit(app.exec\_())

**if** \_\_name\_\_ **==** '\_\_main\_\_':

**try**:

        print("Connecting to DB")

        db **=** pymysql.connect("localhost", "root", "", "drug\_inventory")

        cur **=** db.cursor()

        print("Succesfully connected to DB")

        print("Starting program...")

        openMainWindow()

        cur.close()

        db.close()

        print("Exiting program")

**except** pymysql.DatabaseError:

        print('sorry cannot connect to the database')

        sys.exit()

**except**:

        print ('Exiting program')

1. **SUMMARY AND DISCUSSIONLibraries**

*from* PyQt5 *import* QtCore, QtGui, QtWidgets

This library is used for creating the GUI for the program.

*import* sys

This library is only used several times and that is to exit the program and when the X button is pressed on the GUI and when the program cannot connect to the database.

*import* pymysql

This library is used for the input and output for the database (php my admin / maria DB)

*import* random

This library is used to randomize the item id number.

*import* ast

Abstract Syntax Tree is used to evaluate the data from the data base.

**Classes & Methods**

class Ui\_MainWindow

**def** MyConverter() #converts the data from the database into a tuple

**def** addTable() #add the item into the main table

**def** editItemClicked() #this will show the edit item ui

**def** removeItemClicked() #this will show the remove item ui

**def** refreshClicked() # this will be the trigger to refresh the data

**def** refreshData() #this will refresh the data

**def** setupUi()#this will setup the main window

class Ui\_deleteItem

**def** searchClicked() #this method will search the database for the item id that the user is searching

**def** okClicked() #this will be locked until the search result is non-existent. this will be the function that removes the item in the database

**def** setupUi() #this setup the delete item ui

class Ui\_addItem

**def** okClicked() #this method will be grabbing the text in the input of the user then inserting it into our database

**def** setupUi() #setup the add item ui

class Ui\_editItem

**def** searchClicked() # this method will activate when the search is pressed and it will search the database for the searched item\_id

**def** okClicked() # this method will be activated when the ok button is clicked and in this methon the program will collect the data that you have changed and update the database to make the edit.

**def** setupUi() # this method will setup the ui for the edit window

**Program**

Our group has developed a network of inventory for drugs. The program would benefit people, especially in the healthcare sector. The old form in which the inventory is counted is labor-intensive and expensive. Often people are prone to error, and objects are overcounted or undercounted.