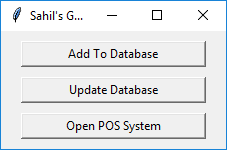
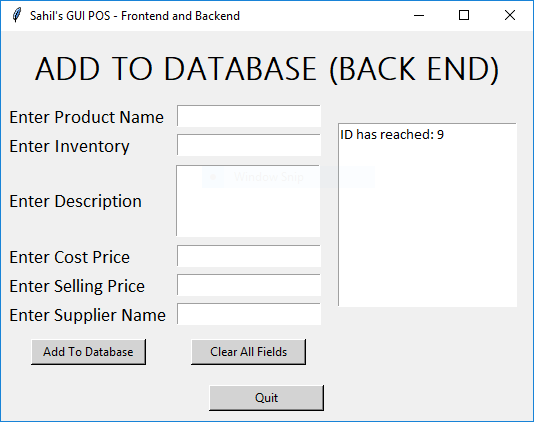
# Startup Window

Startup window displays three buttons to navigate to specific sub-programs, namely:

Startup Window

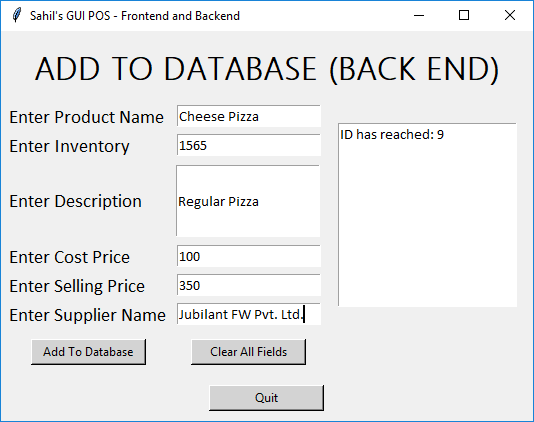
1. Program 1 - Add to Database
2. Program 2 - Update Database
3. Program 3 - Open POS System

# Program 1 - Add to Database

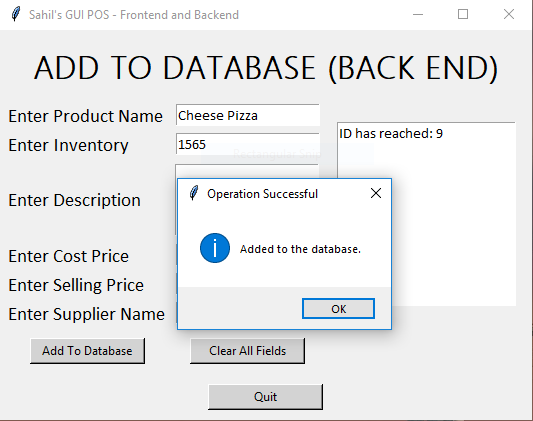


This sub-program adds inserted values into MySQL Database ‘store’ table ‘inventory’ by fetching (get()) required values from entries.

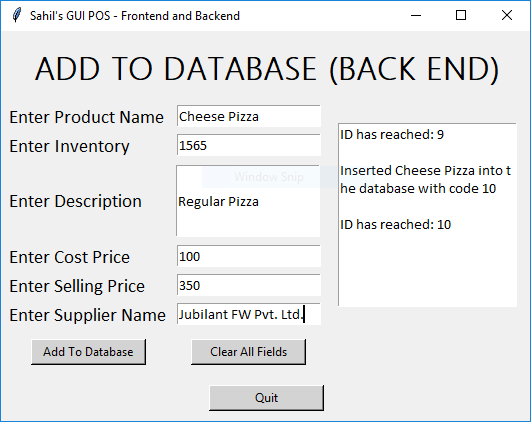
\*Highlighted ID value reflects the last Primary Key (ID) from the ‘inventory’ table.



Values can be inserted in the following way.

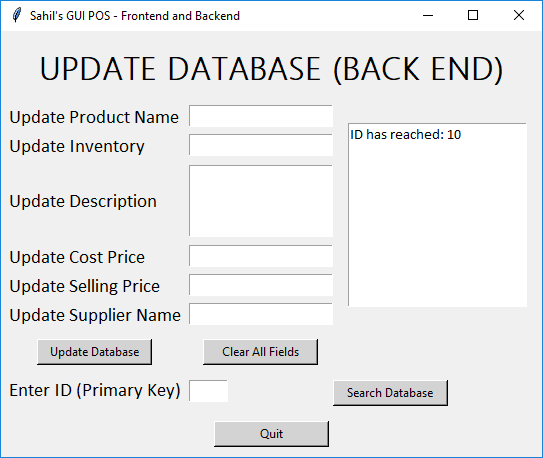


*\*Highlighted: tkinter.messagebox* - UI Element

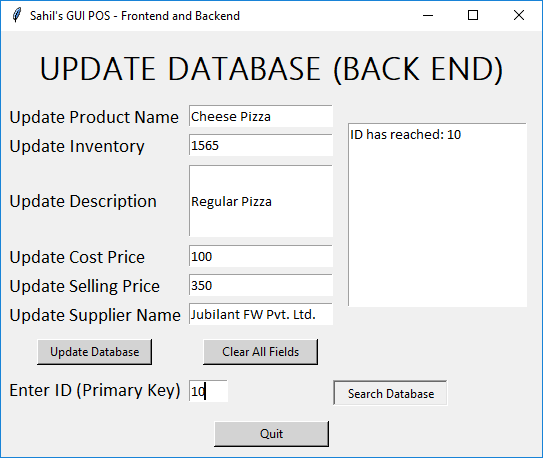


\*Highlighted: Informing user that ID in ‘inventory’ table has reached value 10 and that product has been successfully inserted into the database.

# Program 2 - Update Database



This sub-program fetches values from MySQL database *store.inventory* and displays them in the given entries to be updated.

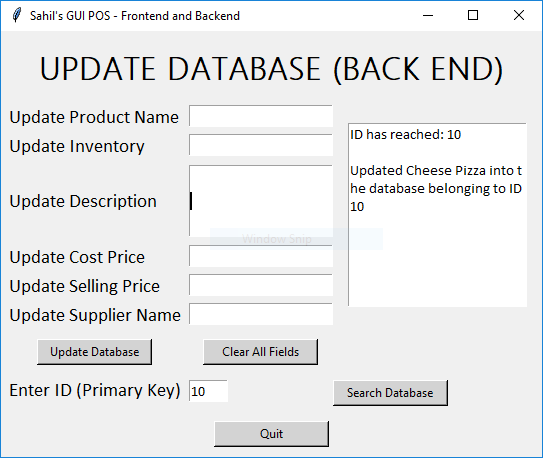
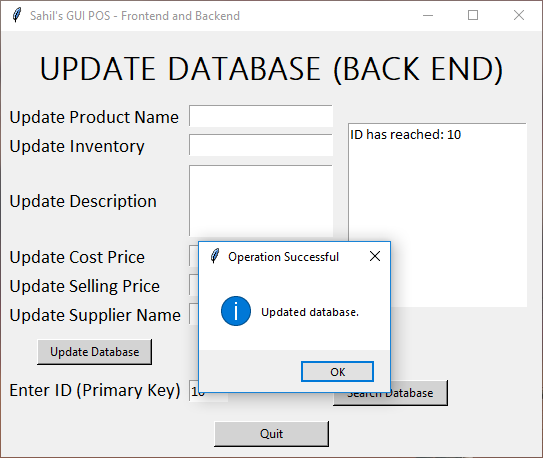


Database searched for Primary Key ‘10’ **(as was added using Program 1)** and values were reflected correctly as shown.



Here description for ‘Cheese Pizza’ was changed from ‘Regular Pizza’ to ‘Cheese Pizza’, and stock quantities were updated, as is reflected in the MySQL Query.



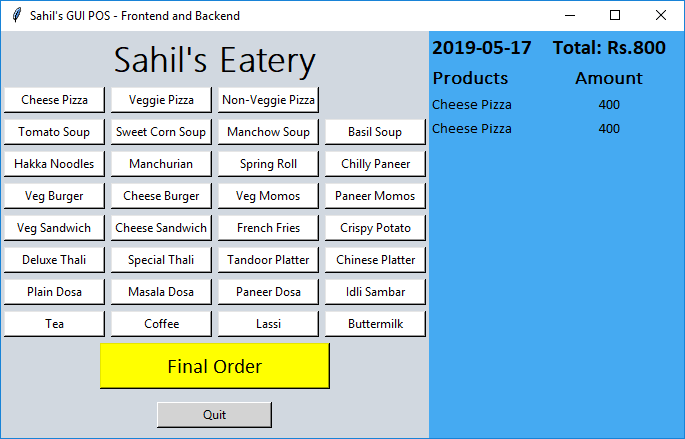


*\*Highlighted: tkinter.messagebox* - UI Element & Textbox updated for UI

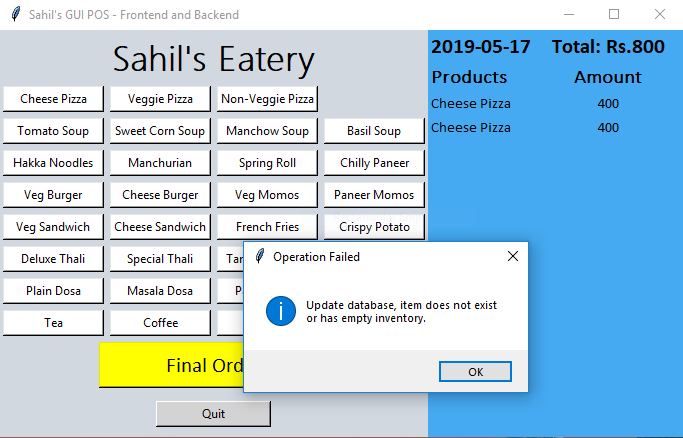
# Program 3 - POS System

\*Highlighted: Order buttons for ease of use and speed

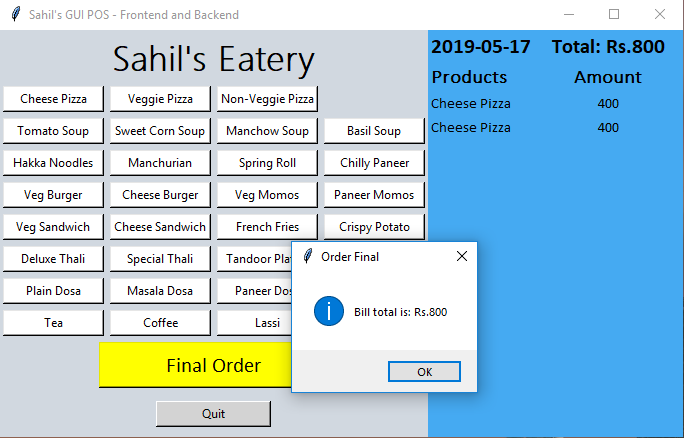
This sub-program is the main feature, it creates invoices, calculates change, updates the ‘transactions’ database, updates inventory from ‘inventory’ table and creates a text invoice.



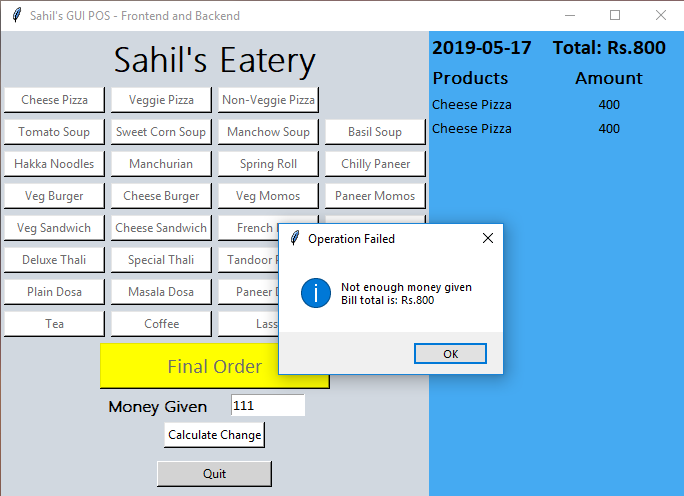
\*Highlighted: Added items shown in right frame along with total order amount



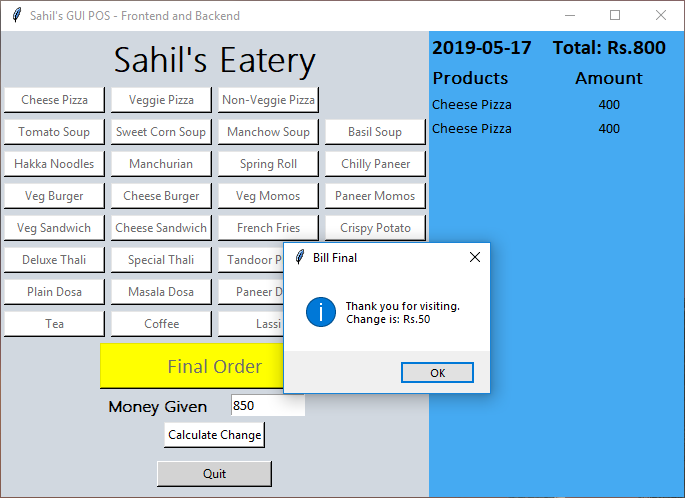
If order button pressed and item does not exist, or has zero stocks, message is displayed.



*Final Order* button pressed, *tkinter.messagebox* displayed and order buttons are **DISABLED**



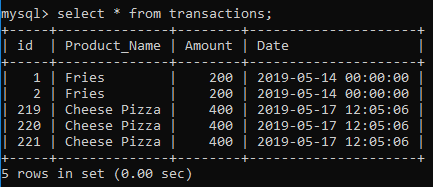
\*Highlighted: As *Final Order* pressed, all order buttons disable and fade, dynamically showing change calculator, if money given lower than order, *tkinter.messagebox* displayed saying money not enough.



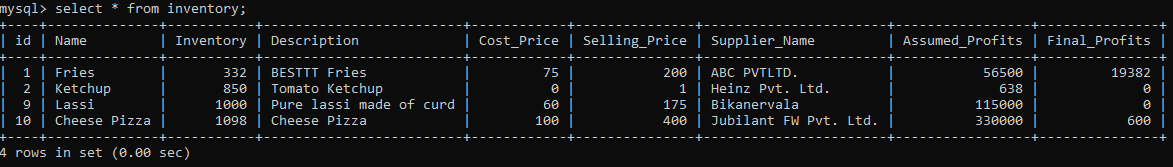
\*Highlighted: Given amount more than or equal to Total Bill, *tkinter.messagebox* displays change requires, order is now final.

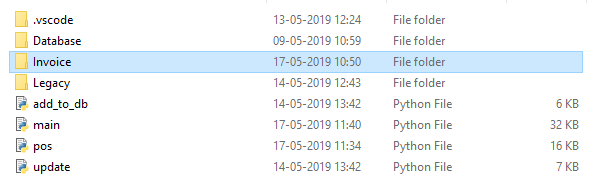


\*Highlighted: Changes reflected in table ‘store.inventory’

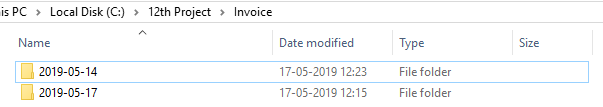
Inventory and Transaction tables are as follows:

Transactions table also shows time and date of item purchase.

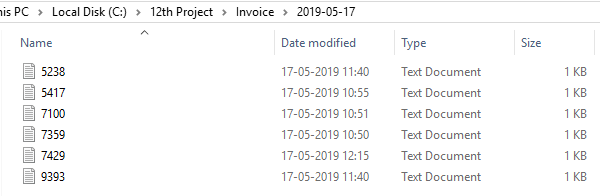
Inventory table updates real profits as all items may not be sold and compares them to assumed profits (if all items were sold)



For program *main.py* dynamically invoices are stored in folder at same location *Invoice* using concepts of file creation.



Invoices in *Invoice* folder are saved according to date.



A random number (*random.randrange(5000, 10000)*)used to generate random invoices.

Here is the invoice generated for the sample example used in this output.

Note that even if item name is long, the *Amount* and *S.No.* panels will not shift to the right or left if the name is small.

(*(self.name + ‘ ‘)[:14]* used which prevents this)

