



UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

# Coffee Management System Report

Nguyen Hoang Nam BI10-123

Nguyen Quoc Hung BI10-072

Nguyen Minh Duc BI10-031

Luong Nguyen Viet Son BI10-156

Nguyen Tien Dat BI10-028

*2020-2021*

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	What does it do? . . . . .	2
1.2	Advantages of this project . . . . .	2
<b>2</b>	<b>How we created it</b>	<b>3</b>
2.1	Database schema . . . . .	3
2.2	Python modules, classes, input, output . . . . .	4
2.3	UI Structure . . . . .	6

---

# 1 Introduction

## 1.1 What does it do?

These are the main functions of this app :

### Input the data

The user can input the required data for each database table. Via button showing up on the main window, Users write the value into each blank space box. When they are done they can click on the ADD button according to each table that they want to add the data. In case of some data that already added to the table, the system will notify the user through a message box that said the added data already available on the system.

### Preserving the data

With the help of a database, the data will be store. We design the save file to be as light as possible so it can be stored for months without making the hard drive full. Therefore, it helps to trace back the bill in case of any trouble occurs. Also, reduce the time for storage checking by matching data across the database. Finally, when it comes to a huge brand of coffee shops like Highland or CoffeeHouse, the manager can copy the data from each store for total checking.

### Manipulating the data

The user can change and switch the data in the database for later use. Also, with the link inside the database, for each change, the data across the database will be reflected immediately, reduce the time of inputting data. In case of incorrect input, the user can manually change the data in the database freely. We make an edit module that able to cross-checking with the database to preserve the data consistency.

### Getting the data

The user can get the data from each table out for checking and calculating. For each table, we set up a searching module, that helps the user getting the data from the database. The searching module has various options based on the table, which help the user can get exactly the data they want to get.

## 1.2 Advantages of this project

We are confident to say that with the following advantages, users will choose our coffee shop management system than the others

### Easy to use

This coffeeshop management system is simplify designed for the users. Each feature is designed with a separate area, a blank space to fill the data, and a button corresponding to each function. Therefore, users can easily learn how to use it with no hard time finding the manual.

### **Back up and recovery**

With a designated database, this management system can able to create and store data in a file. The user can decide how to store this data in many ways and forms. You can pack it into a zip file, push it into the internet cloud system or store it in a USB for later use. Therefore, minimize the chance of losing data and help recover the data.

### **Privacy**

This system is able to work without the internet. Thus, it impossible to DDoS this system, make it more sustainable. Also, this application works individually on each device used, therefore, only authorized devices can use this app.

### **Data consistency**

With a link in each table we created in this management system, the data will be matched across the database. That makes all the data in the database appear consistent and stay the same for user viewing. Each change in the database reflected immediately so there is no data redundancy.

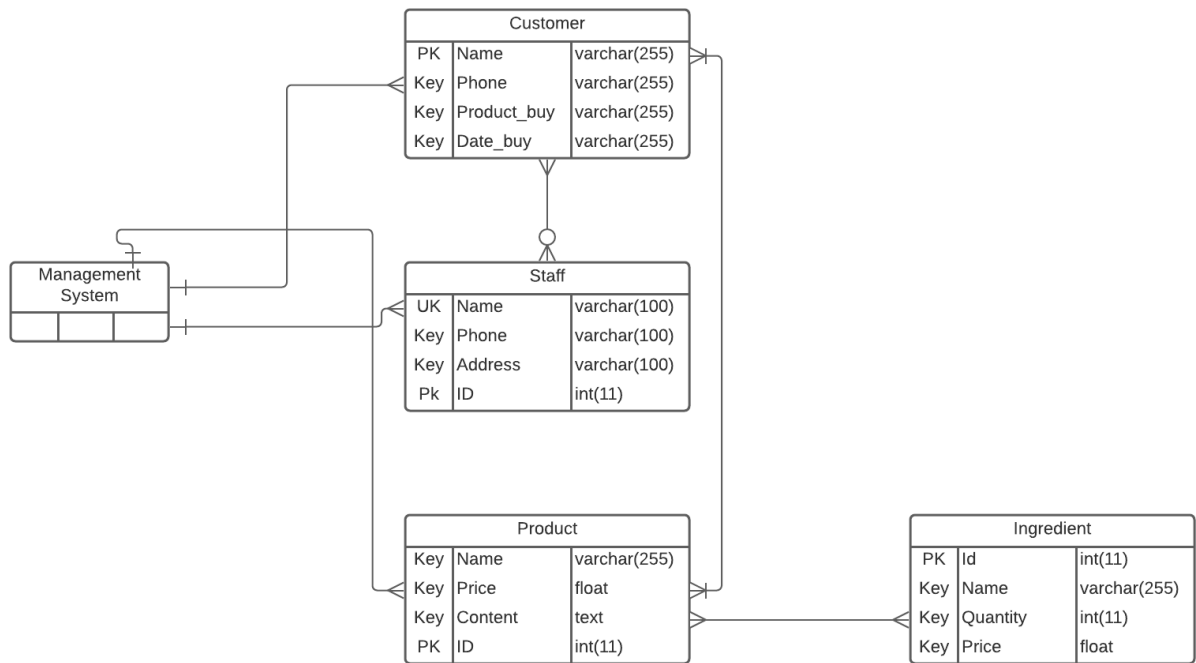
### **Apdation**

The database can easily change to adapt to the requirement. So for each coffee shop, the owner can set up their own database for use. Hence this management system works flexibly in all situations and user requirements.

## **2 How we created it**

### **2.1 Database schema**

- To be able to keep information and make an easy way to access, we created a database called “coffee\_management”
- There are 4 tables: Product, Staff, Customer, Ingredients
- Here is the database schema



The relationships between Customer and Product, Product and Ingredient are many to many. It means a customer can buy many products and a product can be purchased by many customers.

## 2.2 Python modules, classes, input, output

### Modules

- Customer.py implements ' class Customer '
- Product.py implements ' class Product '
- Staff.py implements ' class Staff '
- TableCustomer.py, TableProduct.py, TableStaff.py : create tables that will be shown when we run Show.py
- ingwindow.py : implements button and open another window to edit and search for ingredients
- Validation.py : it shows messages to inform us what happened after we execute a certain command
- Globalvariable : implements global variable
- Error.py : shows message boxes when it spots an error
- Show.py : imports these above modules to start the program

### **Class Customer**

- def ADDCUSTOMER() : function that adds customer to customer table, and will be executed by using 'Add Customer' button

- def DELETECS() : function that deletes customer from customer table, and will be executed by using 'Delete Customer' button

- def EDITCUSTOMER() : function that changes an existing customer's information in customer table, and will be executed by using 'Edit Customer' button

- def search\_customer() : function that open another window with a table and an empty bar to type in who we want to find. The table inside of it shows the results that match

- def searchcustomer() : function that search for every available results. Executed by using 'Search Customer' button

### **Class Product**

- def ADDPRODUCT() : function that adds product to product table, and will be executed by using 'Add Product' button

- def DELETEPRODUCT() : function that deletes product from product table, and will be executed by using 'Delete Product' button

- def EDITPRODUCT() : function that changes an existing product's information in product table, and will be executed by using 'Edit Product' button

- def search\_product() : function that open another window with a table and an empty bar to type in what we want to find. The table inside of it shows the results that match

- def search() : function that search for every available results. Executed by using 'Search Product' button

### **Class Staff**

- def ADDSTAFF() : function that adds staff to staff table, and will be executed by using 'Add Staff' button

- def DELETESTAFF() : function that deletes staff from staff table, and will be executed by using 'Delete Staff' button

- def EDITSTAFF() : function that changes an existing staff's information in product table, and will be executed by using 'Edit Staff' button

- def search\_staff() : function that open another window with a table and an empty bar to type in who we want to find. The table inside of it shows the results that match.

- def searchstaff() : function that search for every available results. Executed by using 'Search Staff' button

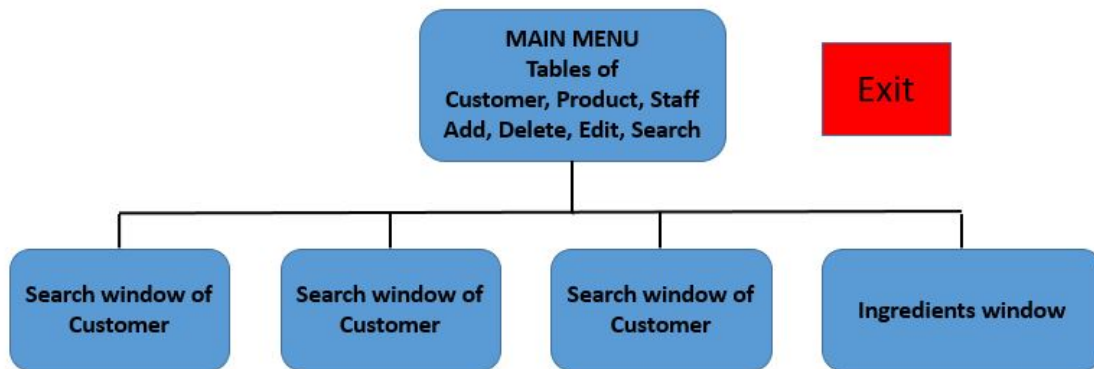
- In 3 classes above, each has def \_\_init\_\_(self): to initialize attributes( such as name, phone, id, of its own class

### Input & Output

- Input: it's the strings that we type in the entry

- Output: after executing order by clicking on the button, the output will be displayed on the table that we created previously.

## 2.3 UI Structure



Main menu

Coffee Management

Product name

Product price

Product content

Staff name

Staff phone

Staff address

Customer name

Customer phone

Product buy

Date buy

Exit

Ingredients

Add Product

Delete Product

Edit Product

Search Product

Add Staff

Delete Staff

Edit Staff

Search Staff

Add Customer

Delete Customer

Edit Customer

Search Customer

ID	Name	Price	Content
2	Cold Coffee	100000.0	Good for health
5	Cold Tea	100000.0	Good for health
7	Hot Milk	10000.0	Good for health
8	Cold Milk	10000.0	Good for health
11	Lipton	10000.0	Good for health
12	Lavie	10000.0	Good for health

ID	Name	Phone	Address
1	Minh Chen	012345678	Quang Binh
2	Nam	012345678	Hà Nội
3	Huy	0936255821	Thanh Hóa
4	Hải	0325968459	Lạng Sơn
5	Dạt	0598632359	Hải Phòng
6	Hùng	0598785441	Đông Tháp

Name	Phone	ProductBuy	DateBuy
Dương	02398631147	Fanta	4/8/2019
Lương	02540951512	Fanta	2/4/2020
Lương An	02305656696	Lavie	5/6/2019
Nam	03233233323	Hot Coffee	2/2/2021
Thái	02398874436	Cold Milk	4/8/2020
Tung	02367459983	Cold Tea	2/7/2019

Ingredient window

Ingredient

Ingredient name

Ingredient quantity

Ingredient price

Add Ingredient

Delete Ingredient

Edit Ingredient

ID	Name	Quantity	Price
1	Sugar	20	200000.0
2	Milk	20	200000.0

Search ingredient

Search By ...

Search ingredient

ID	Name	Quantity	Price
----	------	----------	-------

Search windows

For Customer

Search Customer

Search customer

Search By ...

Search customer

Name	Phone	ProductBuy	DateBuy
------	-------	------------	---------

For Product

Search Product

Search product

Search By ...

Search product

ID	Name	Price	Content
----	------	-------	---------

For Staff

Search Staff

Search staff

Search By ...

Search staff

ID	Name	Phone	Address
----	------	-------	---------