# Cashier Application (POS System) for CRISTY’S LOVE BURGER HUB

Asmit Gahatraj B.K.

B.Sc. (Hons.) Computing, Softwarica College of IT and E-commerce, Coventry University

ST4008CEM: Computing Activity Led Learning Project 1

Giriraj Rawat

July 26, 2022

# TABLE OF CONTENTS

Cashier Application (POS System) for CRISTY’S LOVE BURGER HUB 5

Introduction 5

Employee Database 5

Product Database 7

Billing Table 8

Order Table 9

Register Function 10

Update Function 11

Delete Function 12

Testing 13

Version Control 19

Conclusion 20

# TABLE OF FIGURES

[Figure 1 4](#_Toc109689148)

[Figure 2 5](#_Toc109689149)

[Figure 3 6](#_Toc109689150)

[Figure 4 7](#_Toc109689151)

[Figure 5 8](#_Toc109689152)

[Figure 6 9](#_Toc109689153)

[Figure 6 10](#_Toc109689154)

[Figure 7 11](#_Toc109689155)

[Figure 9 12](#_Toc109689156)

[Figure 1o 13](#_Toc109689157)

[Figure 11 13](#_Toc109689158)

[Figure 12 14](#_Toc109689159)

[Figure 13 15](#_Toc109689160)

[Figure 14 15](#_Toc109689161)

[Figure 15 16](#_Toc109689162)

[Figure 16 16](#_Toc109689163)

[Figure 17 17](#_Toc109689164)

[Figure 18 18](#_Toc109689165)

# Cashier Application (POS System) for CRISTY’S LOVE BURGER HUB

# Introduction

Creating Database and Testing the success of this POS System is the aim of my part. Database uses Sqlite3 and is based on two tier-database. Testing is performed using black box testing.

# Employee Database

It keeps the record for every information that the supervisor inputs in the employee registration. ‘Manager Database’ and ‘Staff database’ are its types. Its attributes are f\_name, l\_name age, gender, pin, re-pin, father\_name, phone, address, city, zip code, employee\_id (manager\_id/staff\_id) and status as shown in figure 1 and 2.

### Figure 1

*Source code of Manager Database*

Text

Description automatically generated

### Figure 2

*Source code of Staff Database*

Text

Description automatically generated with low confidence

# Product Database

Product database contains the record of the product which helps to identify the product. The attributes for this are product\_name, product\_price, product\_id as shown in figure 3.

### Figure 3

*Source code of Product Database*

**Text

Description automatically generated**

# **Billing Table**

Billing Table keeps the records of every bill generated after the order is taken. The attributes in this table are bill\_id, bill\_date and status as shown in figure 4.

### Figure 4

*Source code of Billing Table*

Graphical user interface, application

Description automatically generated

# Order Table

Order Table keeps the record of every single order made by the staff for the customers. The attributes contained in this table are order\_id, product\_name, quantity, price and total\_price in which order\_id is a primary key as shown in figure 5.

### **Figure** 5

*Order Table*

Graphical user interface, text, application

Description automatically generated

# Register Function

This function adds new employees by the superiors in the database as shown in figure 6.

### Figure 6

*Register function*

*Text

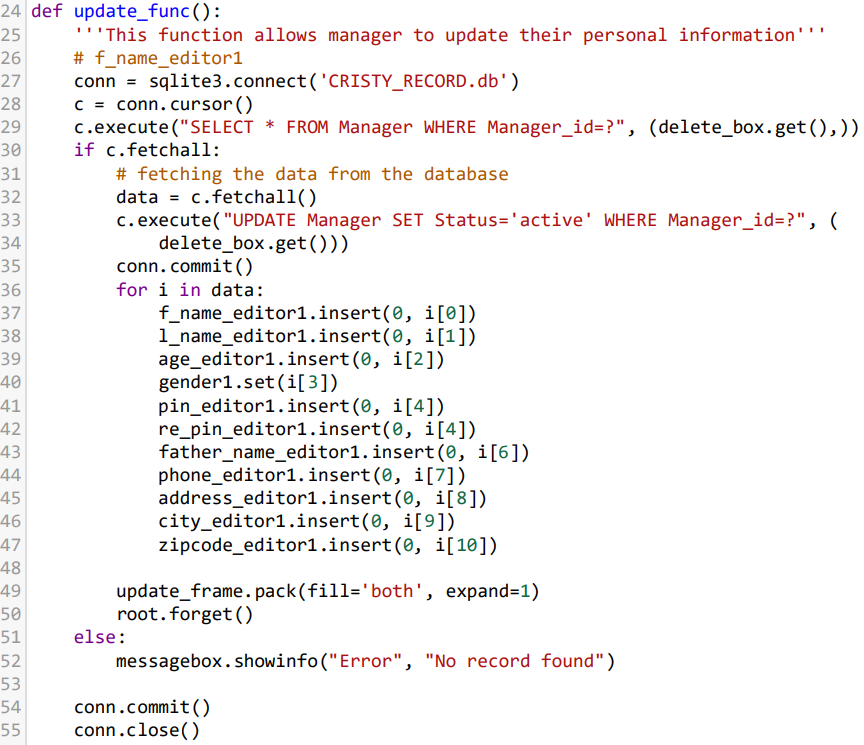
Description automatically generated*

# Update Function

This function is defined for employees to change their individual information as per their needs as shown in figure 6 for manager data modification.

### Figure 6

*Update function*

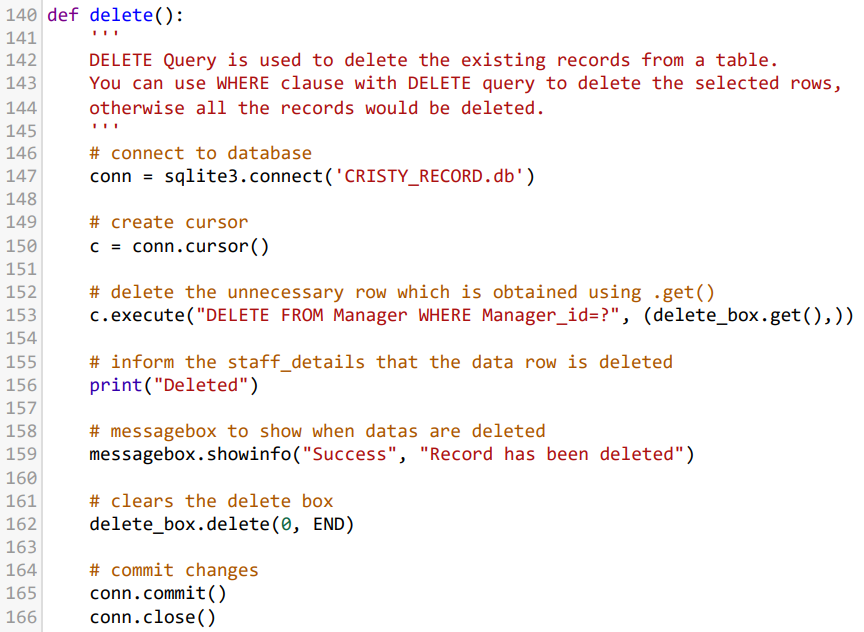


# Delete Function

This function is defined to delete the needless records from any table according to the user’s will especially in case of staff resignation. Superiors can delete anything from the table if it’s not needed anymore as shown in figure 7.

### Figure 7

*Delete function*

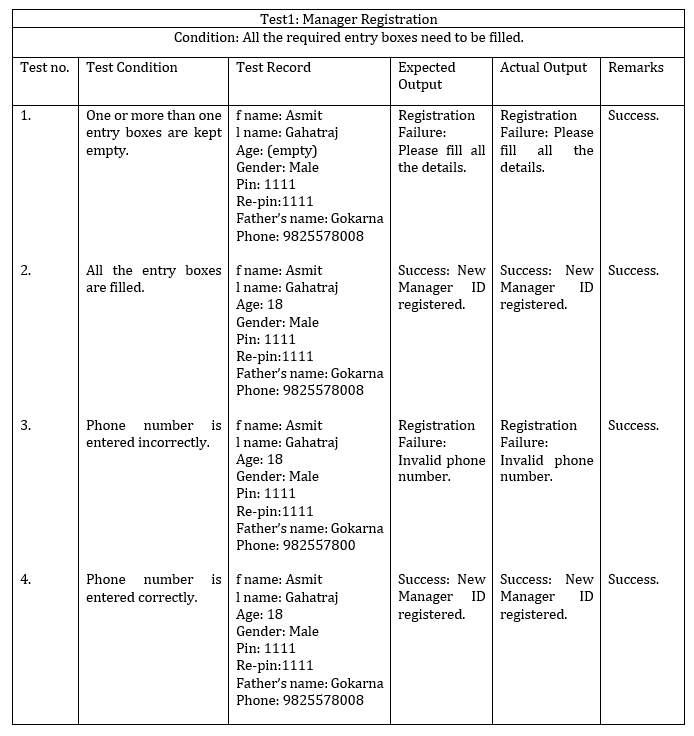


# Testing

System testing was conducted with ‘Black Box Testing’ with results as shown in figures 9-17.

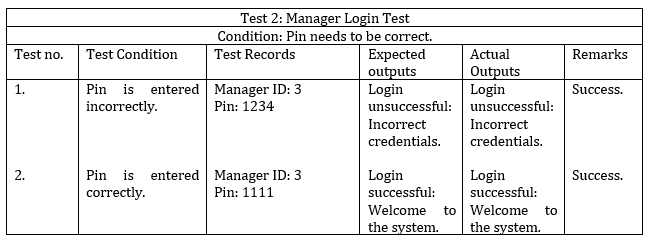
### Figure 9

*Manager registration test*



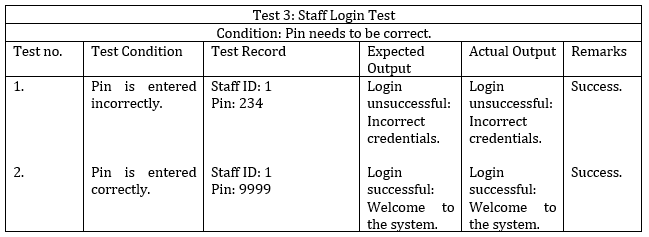
### Figure 1o

*Manager login test*

**

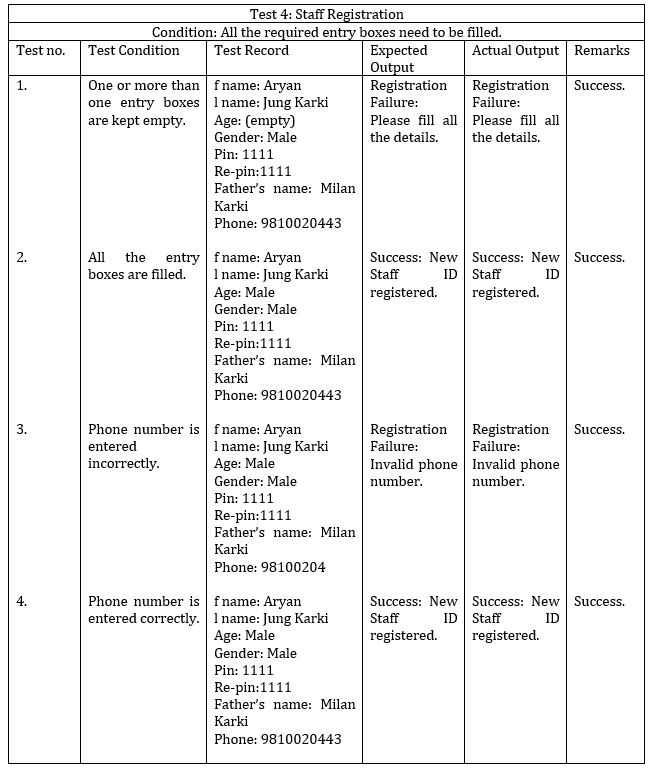
### Figure 11

*Staff Login Test*

**

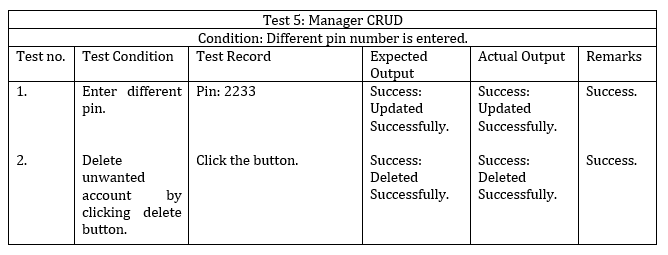
### Figure 12

*Staff registration test*

**

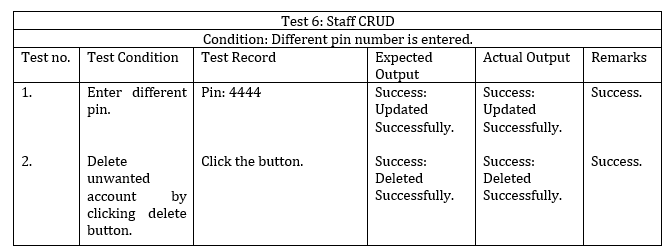
### Figure 13

*Manager CRUD Test*

**

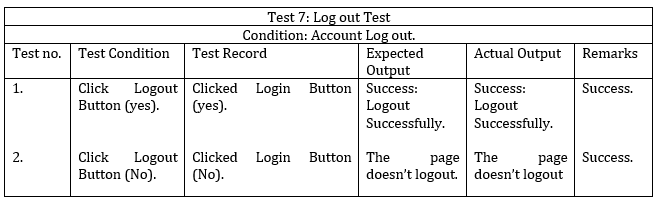
### Figure 14

*Staff CRUD Test*

**

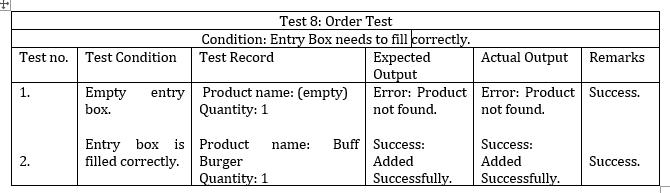
### Figure 15

*Logout Test*

**

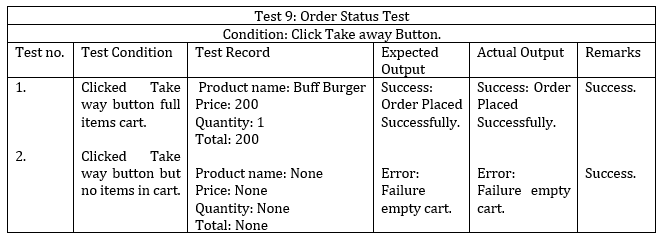
### Figure 16

*Order Test*

**

### Figure 17

*Order Status Test*

**

# Version Control

**Github**:

### Figure 18

*Github commit*

# Conclusion

The goal of my part in this project is to make database using sqlite3 module that records information of registered employees and testing ‘*validity/invalidity’* of the program by using ‘black box testing’ method as taught in our course. The database was based on two-tier architecture. Defining functions related to *Create-Update-Delete* of CRUD was also my part. It felt challenging to create a database and define functions to connect to the GUI as my first project. I actively took part in our team meetings and referenced to study materials to get my job well done. This helped me in my information gathering skills and its proper utilization.

In my future projects, I would experiment with many other types of database structures and choose better alternatives to provide better efficiency and security. Testing taught me that even a working software has many hidden bugs, debunking developer’s myth that a working software has no errors, as taught in classroom .