CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Success in the baking industry comes from making quality products efficiently, while keeping operating costs low. An effective bakery management system tackles your company challenges in one integrated system, meeting your inventory, processing, distribution, and accounting needs. In a fast-paced industry, solutions that improve efficiency and everyday operations are the gold standard for bakery success. Regardless of the size of your operation or the number of locations you have. A bakery management system should be robust enough to handle all the bakery operations, Bakery Management System tackles your industry needs with ease. From handling orders to managing inventory, the Bakery Management System should help you take care of everything.

1.2 OBJECTIVE

The objective of our project is

- To provide a cost-effective Management System.
- Bakeries can have unique id and password for admin users.
- Bakeries can store information about their Employees and customers.
- Bakeries can keep track of product inventory.
- Bakeries can perform daily transactions and generate bills.
- Bakeries can keep track of their transaction history.

1.3 PROPOSED SYSTEM

The proposed system is a console application which provides the bakery administrator / owner to manage the functions of the bakery with ease. Our proposed system consists of the following features

- Each product will be assigned with a unique ID
- For each product a price will be set, along with the quantity based on the availability of the product (i.e., stock of the product) and the products can be stocked in and stocked out as per requirements.

- The stock of a particular product can be checked by specifying the product ID or the stock of all products can be checked at once.
- Each employee will be assigned with a unique employee ID once the Admin enters the employee details into the proposed system.
- The performance of each employee can be tracked using the generated employee ID.
- New customers' details can be entered during transaction and their transactions are tracked using their phone number.
- Our proposed system can get the transaction details of all the customers on particular date or the transaction history of a particular customer.
- Our proposed system can get the details of the last 20 transactions.

1.4 APPLICATION

As our proposed system is lightweight and cost – effective it can used in any bakery irrespective of its size. It can run in the most basic systems, i.e., system with minimal software and hardware requirements. Since, the database technology used in the proposed system is serverless, it can be used by the administrators with minimal technical knowledge. It does not ask the administrators to setup the servers before using the proposed system. It allows the administrators to store and retrieve information about the customers, employees and products with ease. The lack of GUI in the proposed system makes it time efficient. The no-nonsense nature of our proposed system makes it highly efficient in the work place.

CHAPTER 2

HARDWARE AND SOFTWARE REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

Processor: Intel Core 2 Duo E6300 1.86 3.0GHz 6MB 775 Processor & above

RAM: 4 GB

Hard Disk: 1 GB

2.2 SOFTWARE REQUIREMENTS

Operating System: Windows 7-11, Linux-Ubuntu 16.04-17.10, macOS X & above

Language: Python

IDE: Atom text-editor, Command Prompt [Cmd]

Back End: SQLite3

2.2.1 INTRODUCTION TO LANGUAGE

PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. [1]

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.[1]

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts. [1]

2.2.2 INTRODUCTION TO IDE

ATOM

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in JavaScript, and embedded Git Control. Developed by GitHub, Atom is a desktop application built using web technologies. Most of the extending packages have free software licenses and are community-built and maintained. Atom is based on Electron (formerly known as Atom Shell), a framework that enables cross-platform desktop applications using Chromium and Node.js. Atom is written in CoffeeScript and Less, but much of it has been converted to JavaScript.[2]

Atom is a "hackable" text editor, which means it is customizable. There is an init script one can customize using CoffeeScript, a style sheet to customize the looks of Atom, and a keymap to map or re-map key combinations to commands. One can even make a package to wrap all of this functionality into a single package, written in their choice of CoffeeScript or JavaScript.Its developers call it a "hackable text editor for the 21st Century", as it is fully customizable in HTML, CSS, and JavaScript.[2]

COMMAND PROMPT

Cmd is the default command-line interpreter for the OS/2, eComStation, ArcaOS, Microsoft Windows (Windows NT family and Windows CE family), and ReactOS operating systems. The name refers to its executable filename. It is also commonly referred to as Cmd or the Command Prompt, referring to the default window title on Windows.[3]

2.2.3 INTRODUCTION TO BACKEND

SQLite3

SQLite3 is a database engine, written in the C language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded database. It is the most widely deployed database engine, as it is used by several the top web browsers, operating systems, mobile phones, and other embedded systems.[4]

SQLite3 has bindings to many programming languages. It generally follows PostgreSQL syntax but does not enforce type checking. This means that one can, for example, insert a string into a column defined as an integer. Unlike client–server database management systems, the SQLite3 engine has no standalone processes with which the application program communicates. Instead, the SQLite3 library is linked in and thus becomes an integral part of the application program. Linking may be static or dynamic. The application program uses SQLite3's functionality through simple function calls, which reduce latency in database access function calls within a single process are more efficient than inter-process communication. SQLite3 stores the entire database (definitions, tables, indices, and the data itself) as a single cross-platform file on a host machine. It implements this simple design by locking the entire database file during writing. SQLite3 read operations can be multitasked, though writes can only be performed sequentially. Due to the server-less design, SQLite3 applications require less configuration than client–server databases. SQLite3 is called zero-conf, because it does not require service management (such as start-up scripts) or access control based on GRANT and passwords.[4]

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1 ER DIAGRAM

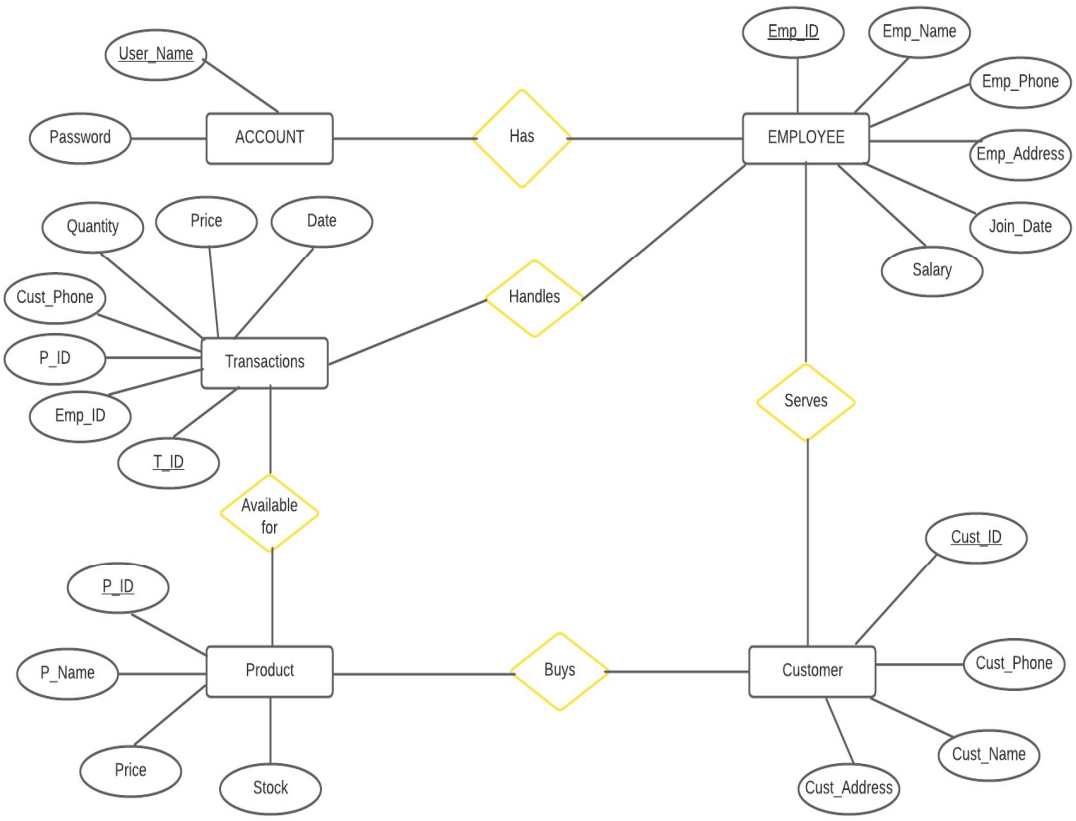


Figure 3.1: ER Diagram of Bakery Management System

In the ER diagram there are 6 tables namely Employee, Customer, Product, Account and Transactions. The Employee table has 1: N cardinality ratio for the relationship (has) to the account table. The Customer table has M: N cardinality ratio for the relationship(serves) to the Employee table. The Product table has N:1 cardinality for the relationship(buys) to the Customer table. The Transaction table has 1: N cardinality ratio for the relationship(available) to the product table. The Transaction table has M: N cardinality ratio for the relationship(handles) to the employee table.

3.2 SCHEMA DIAGRAM

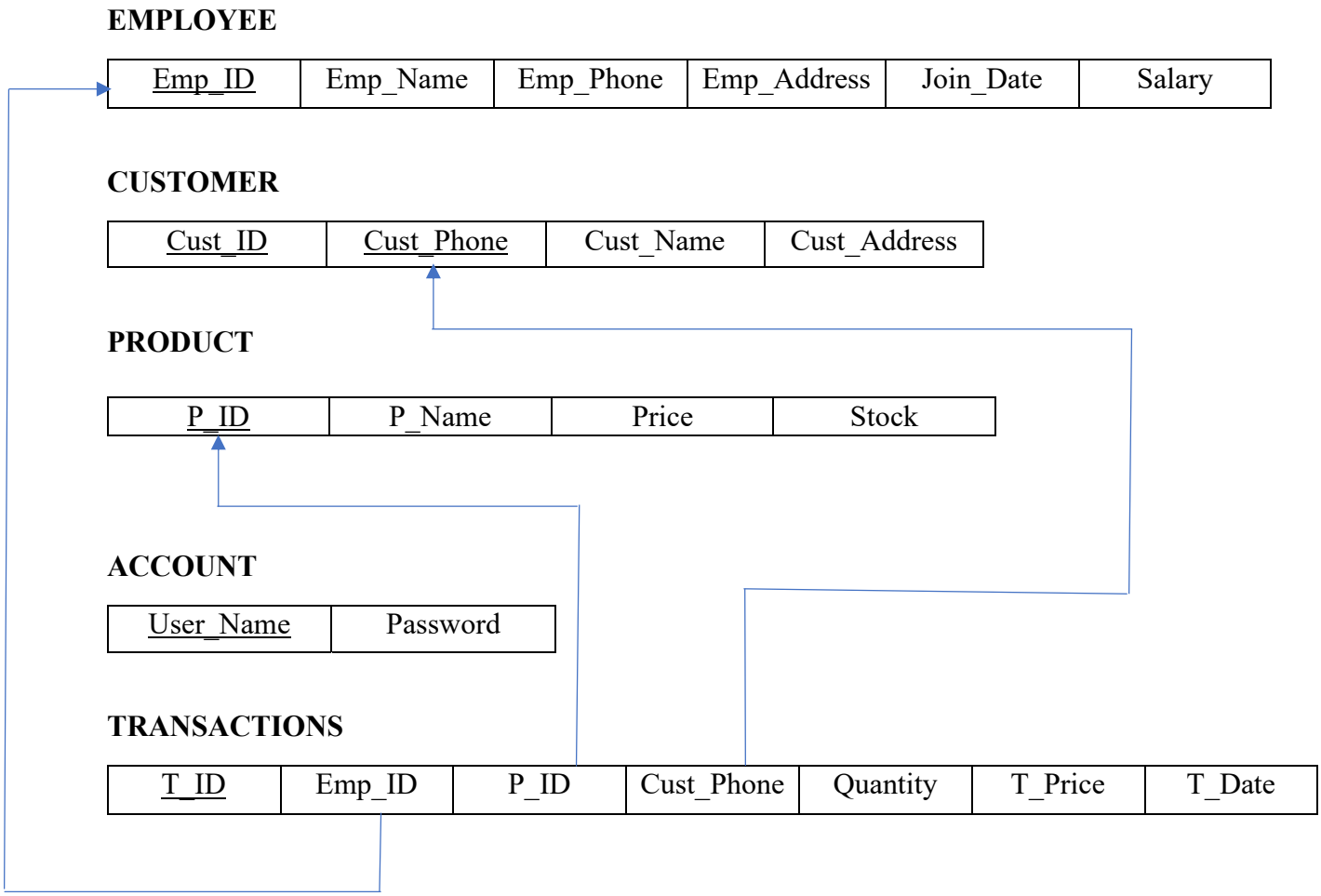


Figure 3.2: Schema Diagram of Bakery Management System

The Schema Diagram consists of 5 tables namely Employee, Customer, Product, Account and Transactions. In the Employee table Emp_ID is the primary key. In the Customer table Cust_ID is the primary key. In the Product table P_ID is the primary key. In the Account table User_name is the primary key. In the Transaction table T_ID is the primary key, it also has foreign keys Emp_ID that references Employee table, P_ID that references Product table and Cust_Phone that references Customer table.

3.3 PSUEDO CODE

```
import sqlite3

conn = sqlite3.connect('bakerydb.sqlite')

cur = conn.cursor()


cur.execute('''

CREATE TABLE IF NOT EXISTS Employee(

Emp_id INTEGER PRIMARY KEY UNIQUE,

Emp_name TEXT,

Emp_Phone TEXT,

Emp_Address TEXT,

Join_date TEXT,

Salary INTEGER

);''') [5]

cur.execute('''

CREATE TABLE IF NOT EXISTS Customer(

Cust_id INTEGER PRIMARY KEY UNIQUE,

Cust_name TEXT,

Cust_Address TEXT,

Cust_Phone TEXT UNIQUE

);''')[5]


cur.execute('''

CREATE TABLE IF NOT EXISTS Product(
```

```
P_id INTEGER PRIMARY KEY UNIQUE,

P_Name TEXT,

Price INTEGER,

Stock INTEGER

); ''')[5]


cur.execute('''

CREATE TABLE IF NOT EXISTS Account(

Username TEXT UNIQUE,

Password TEXT,

PRIMARY KEY(Username)

);''')[5]


cur.execute('''

CREATE TABLE IF NOT EXISTS Transactions(

T_id INTEGER PRIMARY KEY UNIQUE,

Emp_id INTEGER,

Cust_Phone TEXT,

P_id INTEGER,

Quantity INTEGER,

Price INTEGER,

Date TEXT,

FOREIGN KEY(P_id) REFERENCES Product(P_id),

FOREIGN KEY(Cust_Phone) REFERENCES Customer(Cust_Phone),
```

```
FOREIGN KEY(Emp_id) REFERENCES Employee(Emp_id)
);''') [5]


conn.commit()


def StockIn():
    print('Stock-In'.center(211,'-'))
    try:
        productId = int(input('Enter the Product ID:-'))
        quant = int(input('Enter the Quantity to StockIn:-'))
        with conn:
            cur.execute('UPDATE Product SET Stock = Stock + ? WHERE P_id =
?',(quant,productId))
            print('The Stock was updated Successfully')
            cur.execute('SELECT Stock,P_Name FROM Product WHERE P_id = ?',(productId,))
            stock = cur.fetchone()
        print('The Quantity of ',str(stock[1]),'is ',str(stock[0]))
        return
    except:
        print('Wrong product ID , try again')
        return [6]
def AddProduct():
    print('Add a New Product'.center(211,'-'))
    try:
        productName = input('Enter Product Name:-')
```

```
        price = int(input('Enter the Price:-'))
        stock = int(input('Enter the Initial Stock:-'))
        with conn:
            cur.execute('INSERT INTO Product(P_Name,Price,Stock)
VALUES(?,?,?)',(productName,price,stock))
            print('The item was added Successfully')
            cur.execute('SELECT P_id FROM Product WHERE P_Name = ?',(productName,))
            p_id = cur.fetchone()

        print('The Product ID for ',productName,'is ',str(p_id[0]))
       return [6]
    except:
        print('Enter proper Credentials')
        return [6]
def DisplayAll():
    print('Display'.center(211,'-'))
    with conn:
        cur.execute('SELECT * FROM Product')
        dispAll = cur.fetchall()
    print()
    print('%20s %20s %20s %20s'%('Product_ID','Product Name','Price','Stock'))
    print()
    for row in dispAll:
        print('%20s %20s %20s %20s'%(str(row[0]),str(row[1]),str(row[2]),str(row[3])))
    return [6]
```

CHAPTER 4

RESULTS

4.1 SNAPSHOTS

The figure 4.1 shows the login page of the console application where the administrator/ owner / manager can enter their username and password to login.

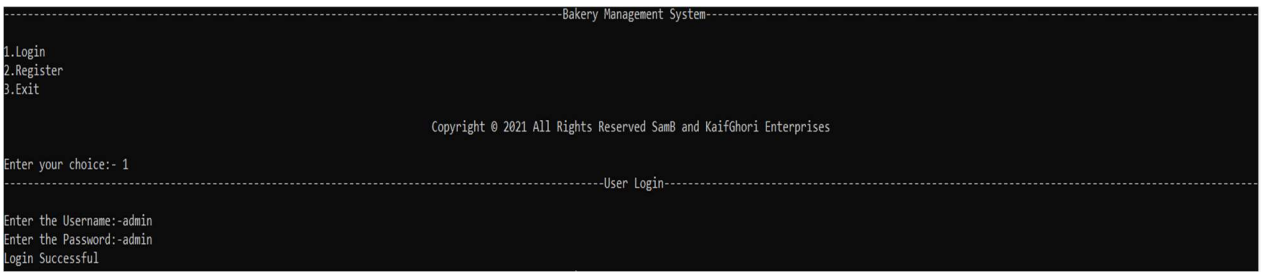


Figure 4.1: Snapshot of login page

The figure 4.2 shows the registration page where new admin users can register themselves and create their own username and password.

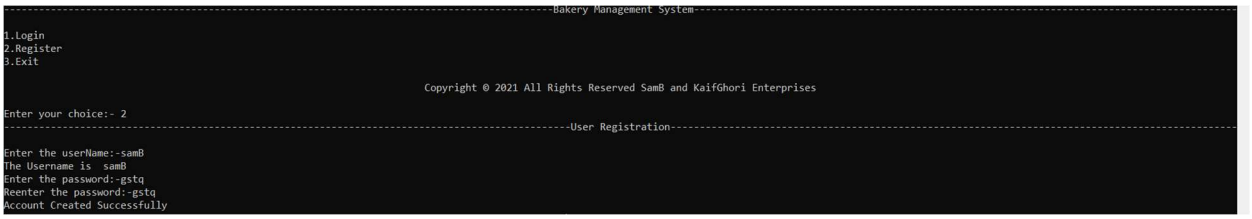


Figure 4.2: Snapshot of the Registration page

The figure 4.3 shows the main menu of the console application from where the admin can either update product, employee, customer information, also provides the option to start a new transaction for a customer and a logout option.

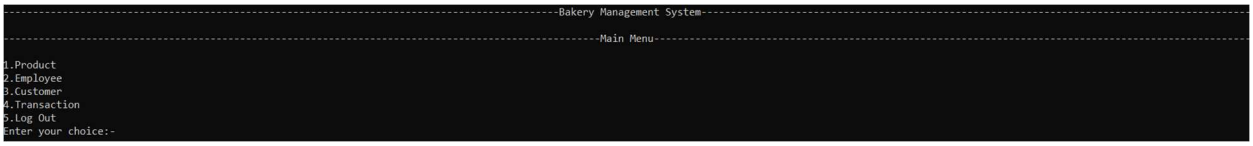


Figure 4.3: Snapshot of the Main menu

The figure 4.4 shows the product menu from where the product stock can be increased and decreased, allows the admin to add new products and provides the option to list all the existing products.

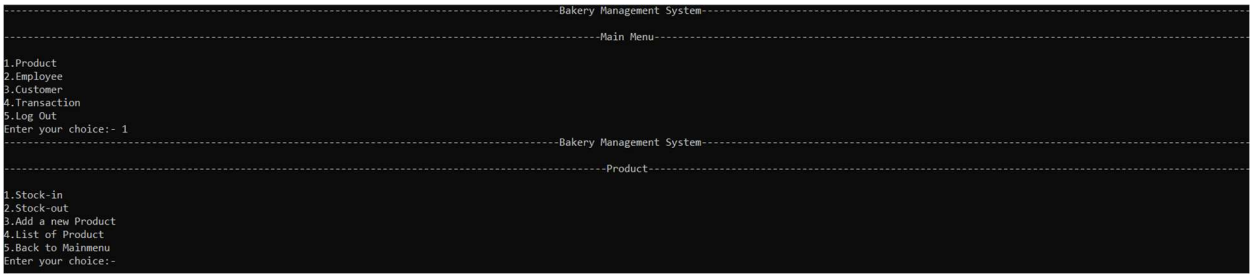


Figure 4.4: Snapshot of the Product Menu

The figure 4.5 shows that once the admin selects the list all products option, he then gets the option to either list all the products or get information about a particular product by entering its product ID.

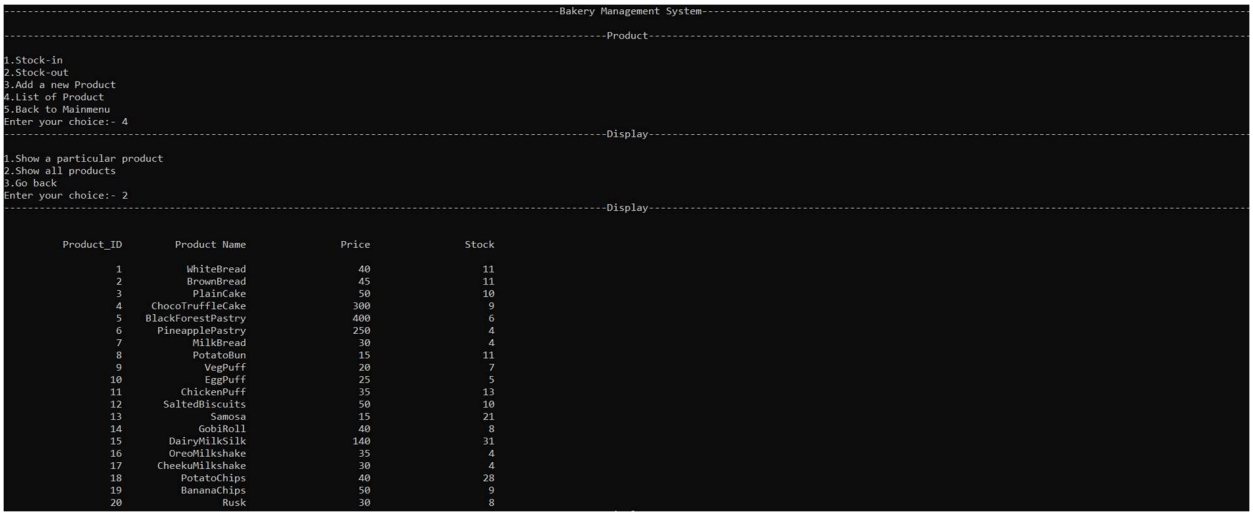
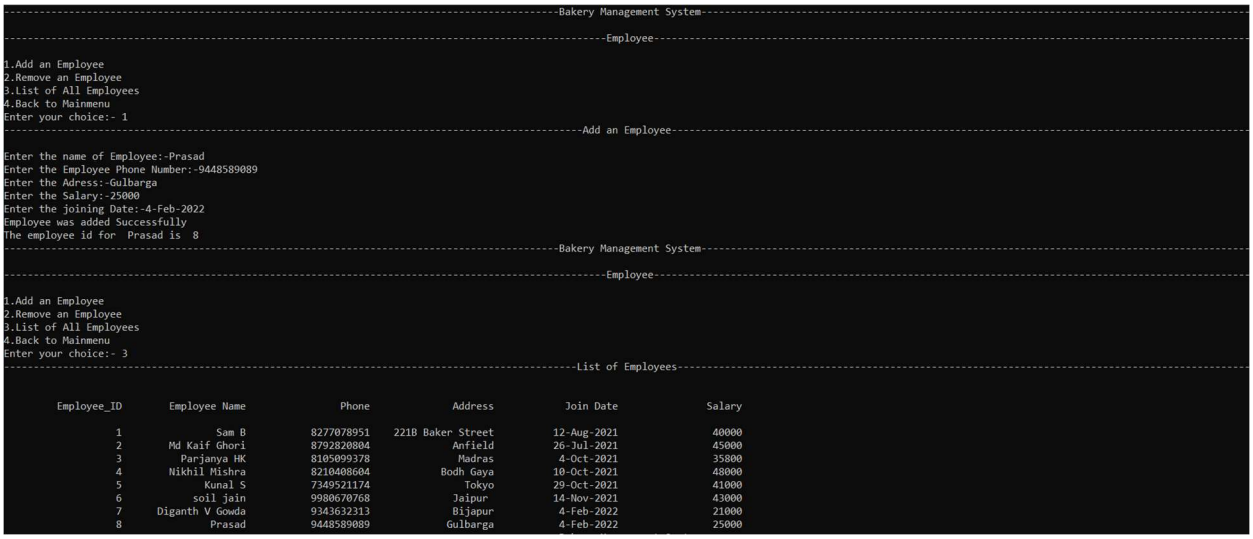


Figure 4.5: Snapshot of the Product Listing

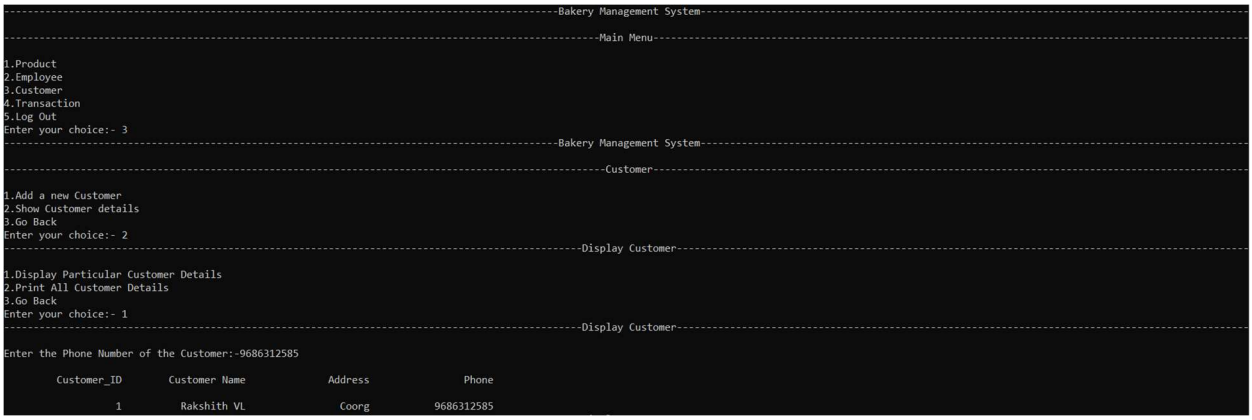
The figure 4.6 shows that when the admin chooses option 2 from the main menu, the admin will be able to add new employees, remove existing employees and be able to get the list of all employees.



```
--------------------------------------------------------------------------------------------------------------Bakery Management System--------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------Employee--------------------------------------------------------------------------------------------------------------

1.Add an Employee
2.Remove an Employee
3.List of All Employees
4.Back to Mainmenu
Enter your choice:- 1
--------------------------------------------------------------------------------------------------------------Add an Employee--------------------------------------------------------------------------------------------------------------

Enter the name of Employee:-Prasad
Enter the Employee Phone Number:-9448589089
Enter the Adress:-Gulbarga
Enter the Salary:-25000
Enter the joining Date:-4-Feb-2022
Employee was added Successfully
The employee id for  Prasad is  8
--------------------------------------------------------------------------------------------------------------Bakery Management System--------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------Employee--------------------------------------------------------------------------------------------------------------

1.Add an Employee
2.Remove an Employee
3.List of All Employees
4.Back to Mainmenu
Enter your choice:- 3
--------------------------------------------------------------------------------------------------------------List of Employees--------------------------------------------------------------------------------------------------------------


        Employee_ID        Employee Name               Phone             Address            Join Date              Salary

                  1                Sam B          8277078951    221B Baker Street          12-Aug-2021               40000
                  2         Md Kaif Ghori          8792820804             Anfield          26-Jul-2021               45000
                  3           Parjanya HK          8105099378              Madras           4-Oct-2021               35800
                  4         Nikhil Mishra          8210408604           Bodh Gaya          10-Oct-2021               48000
                  5              Kunal S           7349521174               Tokyo          29-Oct-2021               41000
                  6            soil jain           9980670768              Jaipur          14-Nov-2021               43000
                  7       Diganth V Gowda          9343632313              Bijapur           4-Feb-2022               21000
                  8                Prasad          9448589089            Gulbarga           4-Feb-2022               25000
```

Figure 4.6: Snapshot of Adding new Employee

The figure 4.7 shows that once the admin chooses option 3 from the main menu, the admin will be able to add new customers, get details of a particular customer or even get the list of all customers and their details.



```
--------------------------------------------------------------------------------------------------------------Bakery Management System--------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------Main Menu--------------------------------------------------------------------------------------------------------------

1.Product
2.Employee
3.Customer
4.Transaction
5.Log Out
Enter your choice:- 3
--------------------------------------------------------------------------------------------------------------Bakery Management System--------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------Customer--------------------------------------------------------------------------------------------------------------

1.Add a new Customer
2.Show Customer details
3.Go Back
Enter your choice:- 2
--------------------------------------------------------------------------------------------------------------Display Customer--------------------------------------------------------------------------------------------------------------

1.Display Particular Customer Details
2.Print All Customer Details
3.Go Back
Enter your choice:- 1
--------------------------------------------------------------------------------------------------------------Display Customer--------------------------------------------------------------------------------------------------------------

Enter the Phone Number of the Customer:-9686312585

        Customer_ID        Customer Name             Address              Phone

                  1         Rakshith VL               Coorg          9686312585
```

Figure 4.7: Snapshot of accessing particular customer details

The figure 4.8 shows that the admin is accessing the lsit of all the customers and their details.

```
--------------------------------Display Customer--------------------------------

1.Display Particular Customer Details
2.Print All Customer Details
3.Go Back
Enter your choice:- 2
--------------------------------Display Customer--------------------------------


      Customer_ID       Customer Name            Address            Phone

                1        Rakshith VL               Coorg       9686312585
                2     Karpaga Murthy              Mysore       6364574484
                3            Manas A              Mysore       6364381389
                4          Deepakshi              Mysore       9535642779
                5        Karthik B M              Mysore       9110650729
                6            Punya G              Mysore       9972574551
                7    Krishna Talreja                Pune       8095261344
                8    Party Boy Pavan           Hyderabad       6303441339
                9         Manideep G             Bijapur       9742877979
               10          Manaswi M              Mysore       8296564195
               11        Manu Prasad              Mysore       7975562733
               12      Milind S Bhat           Bengaluru       9148415060
               13           Md Afnan               Dubai       7760433057
               14           Md Zaidu              Mysore       7019035142
               15    Md Psudias Khan                Doha       7349724060
               16           Moulya M              Mysore       9379979995
               17        Neha Balaji             Chicago       8317497245
               18         Pratheek N              Mandya       8217415973
               19            Rohan A               Coorg       9019210166
               20          Sacheth K          Trivandrum       9526611134
               21     Preetham Gowda              Hassan       8217656687
               22              Kavan               Coorg       9141172522
```

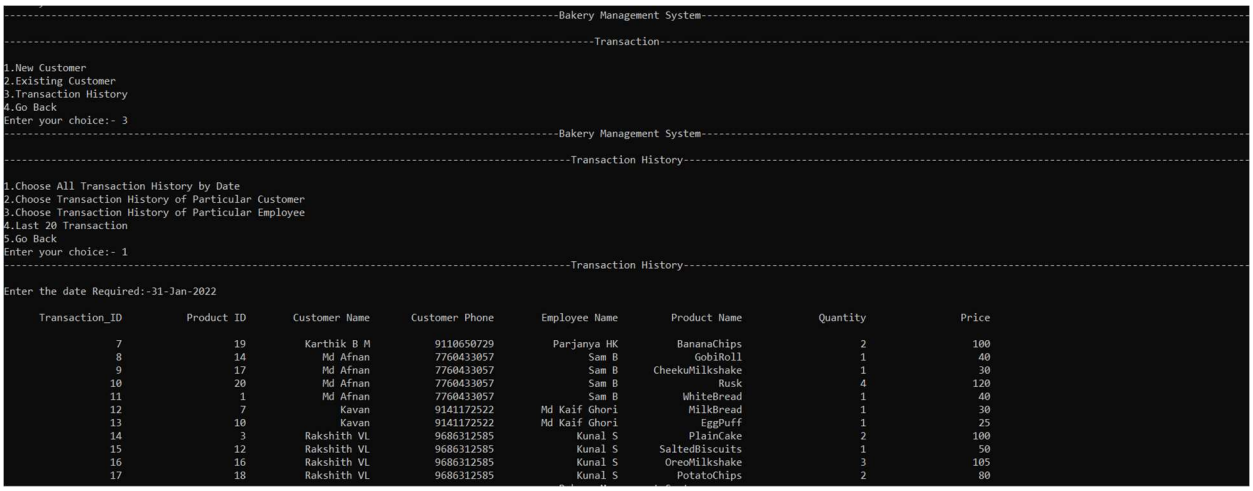
Figure 4.8: Snapshot of Listing all the customer details

The figure 4.9 shows that the admin has initiated a transaction, where first he enters the phone number of the customer and then followed by the product ID that the customer buys, along with the quantity, so that the total amount may be calculated and displayed.

```
-----------------------------Bakery Management System-----------------------------
-----------------------------------Main Menu---------------------------------------

1.Product
2.Employee
3.Customer
4.Transaction
5.Log Out
Enter your choice:- 4
-----------------------------Bakery Management System-----------------------------
----------------------------------Transaction--------------------------------------

1.New Customer
2.Existing Customer
3.Transaction History
4.Go Back
Enter your choice:- 2
-----------------------------------Purchase----------------------------------------

Enter the Phone Number:-8095261344
Enter the employeeId:-3
Enter the Date:-3-Feb-2022
Enter the product id:-3
Enter the quantity:-2
The price of  2 PlainCake is  100
The total Price of the session is  100
Enter the product id:-6
Enter the quantity:-4
Insufficent Stock
Enter the product id:-6
Enter the quantity:-1
The price of  1 PineapplePastry is  250
The total Price of the session is  350
Enter the product id:-
```

Figure 4.9: Snapshot of Transaction in progress

The figure 4.10 shows that the admin can get details about the number of transactions that took place on a particular date.

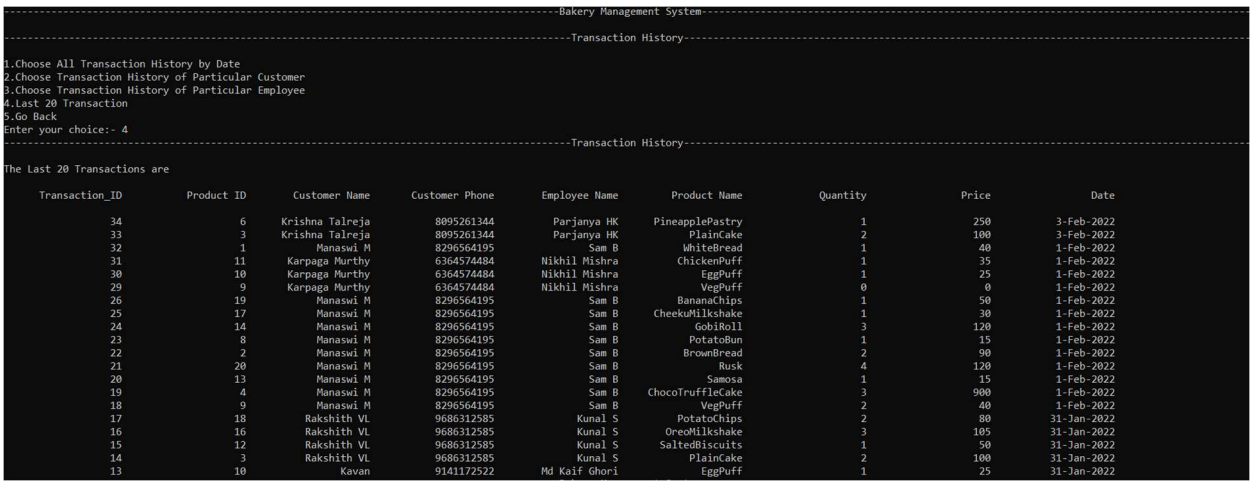


```
----------Bakery Management System----------

----------Transaction----------

1.New Customer
2.Existing Customer
3.Transaction History
4.Go Back
Enter your choice:- 3
----------Bakery Management System----------

----------Transaction History----------

1.Choose All Transaction History by Date
2.Choose Transaction History of Particular Customer
3.Choose Transaction History of Particular Employee
4.Last 20 Transaction
5.Go Back
Enter your choice:- 1
----------Transaction History----------

Enter the date Required:-31-Jan-2022
```

Transaction_ID	Product ID	Customer Name	Customer Phone	Employee Name	Product Name	Quantity	Price
7	19	Karthik B M	9110650729	Parjanya HK	BananaChips	2	100
8	14	Md Afnan	7760433057	Sam B	GobiRoll	1	40
9	17	Md Afnan	7760433057	Sam B	CheekuMilkshake	1	30
10	20	Md Afnan	7760433057	Sam B	Rusk	4	120
11	1	Md Afnan	7760433057	Sam B	WhiteBread	1	40
12	7	Kavan	9141172522	Md Kaif Ghori	MilkBread	1	30
13	10	Kavan	9141172522	Md Kaif Ghori	EggPuff	1	25
14	3	Rakshith VL	9686312585	Kunal S	PlainCake	2	100
15	12	Rakshith VL	9686312585	Kunal S	SaltedBiscuits	1	50
16	16	Rakshith VL	9686312585	Kunal S	OreoMilkshake	3	105
17	18	Rakshith VL	9686312585	Kunal S	PotatoChips	2	80

Figure 4.10: Snapshot of transaction details on particular date

The figure 4.11 shows that the admin can get details about the last 20 transactions that took place recently, also give the option to check the transaction history a particular cuistomer and the number of trasactions performed by a particular employee.



```
----------Bakery Management System----------

----------Transaction History----------

1.Choose All Transaction History by Date
2.Choose Transaction History of Particular Customer
3.Choose Transaction History of Particular Employee
4.Last 20 Transaction
5.Go Back
Enter your choice:- 4
----------Transaction History----------

The Last 20 Transactions are
```

Transaction_ID	Product ID	Customer Name	Customer Phone	Employee Name	Product Name	Quantity	Price	Date
34	6	Krishna Talreja	8095261344	Parjanya HK	PineapplePastry	1	250	3-Feb-2022
33	3	Krishna Talreja	8095261344	Parjanya HK	PlainCake	2	100	3-Feb-2022
32	1	Manaswi M	8296564195	Sam B	WhiteBread	1	40	1-Feb-2022
31	11	Karpaga Murthy	6364574484	Nikhil Mishra	ChickenPuff	1	35	1-Feb-2022
30	10	Karpaga Murthy	6364574484	Nikhil Mishra	EggPuff	1	25	1-Feb-2022
29	9	Karpaga Murthy	6364574484	Nikhil Mishra	VegPuff	0	0	1-Feb-2022
26	19	Manaswi M	8296564195	Sam B	BananaChips	1	50	1-Feb-2022
25	17	Manaswi M	8296564195	Sam B	CheekuMilkshake	1	30	1-Feb-2022
24	14	Manaswi M	8296564195	Sam B	GobiRoll	3	120	1-Feb-2022
23	8	Manaswi M	8296564195	Sam B	PotatoBun	1	15	1-Feb-2022
22	2	Manaswi M	8296564195	Sam B	BrownBread	2	90	1-Feb-2022
21	20	Manaswi M	8296564195	Sam B	Rusk	4	120	1-Feb-2022
20	13	Manaswi M	8296564195	Sam B	Samosa	1	15	1-Feb-2022
19	4	Manaswi M	8296564195	Sam B	ChocoTruffleCake	3	900	1-Feb-2022
18	9	Manaswi M	8296564195	Sam B	VegPuff	2	40	1-Feb-2022
17	18	Rakshith VL	9686312585	Kunal S	PotatoChips	2	80	31-Jan-2022
16	16	Rakshith VL	9686312585	Kunal S	OreoMilkshake	3	105	31-Jan-2022
15	12	Rakshith VL	9686312585	Kunal S	SaltedBiscuits	1	50	31-Jan-2022
14	3	Rakshith VL	9686312585	Kunal S	PlainCake	2	100	31-Jan-2022
13	10	Kavan	9141172522	Md Kaif Ghori	EggPuff	1	25	31-Jan-2022

Figure 4.11: Snapshot of Listing last 20 transaction

CONCLUSION

This project is an attempt to make an efficient way to manage the day-to-day activities of a bakery in action. It is lightweight, easy, convenient and cost-effective for the Bakeries to use. It helps in keeping inventory of the products, the stock of a particular product can be checked by specifying the product ID and the stock of all products can be checked at once, maintain transaction history of customers, can get the transaction details of all the customers on particular date and the transaction history of a particular customer and keep track of sales done by every employee, the performance of each employee can be tracked using the employee ID.

FUTURE ENHANCEMENT

Future enhancement would be to add an interactive GUI to the project. we plan to make it into a fully functional desktop application which can even be used by bigger bakeries. We also plan to extend this project and convert it into an online web application with additional functionalities such as connection between multiple branches of a bakery, and to add the ability for customers to order online.

REFERENCES

[1] Introduction to Language: https://en.wikipedia.org/wiki/Python_(programming_language)

[2] Introduction to IDE: https://en.wikipedia.org/wiki/Atom_(text_editor)

[3] Introduction to IDE: https://en.wikipedia.org/wiki/Cmd.exe

[4] Introduction to Backend: https://en.wikipedia.org/wiki/SQLite

[5] Pseudo code: https://www.tutorialspoint.com/sqlite/sqlite_python.html

[6] Pseudo code: https://docs.python.org/3/library/sqlite3.html