

Bookstore Management System

J COMPONENT PROJECT REPORT

REVIEW 2

Submitted by

ARUNAVA BANERJEE (18BCE2361)

RACHIT TRILOK (18BCE0873)

BINAMRA NEUPANE (18BCE2401)

in partial fulfilment for the award of the degree of

B.Tech

in

Computer Science and Engineering



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

October, 2019

TITLE OF THE PROJECT – BOOKSTORE MANAGEMENT SYSTEM

ABSTRACT

Nowadays, it has become difficult to maintain records and transactions manually for any store. Software systems like our proposed system in the project helps maintain records and transactions. It contains the all information of the book and customer to whom the book is sold. It provides information of the books, customer and the employees of the workshop in a systematic way. The time required to obtain the information is much faster and the information is available in systematic manner. There are a lot of duplicate works and chance of mistake when the records are changed or updated are pretty high. This bookstore management system is made to overcome problems they are facing currently and going for complete transfer from manual systems to computerized systems.

In this project we look over bookstore database management system. What we have tried to do is create an environment using python (tkinter) which gives a GUI that our users can use to store their database upon. We are storing the information of the books available in the bookstore, information of the customer buying from the bookstore and the information of the employees working at the bookstore (booksellers).

The software(s) that we use extensively for this project includes python for creating a window environment and Sqlite for maintaining database.

Basically, the project explores through the basic idea of a Data Base Management System. Applying this very concept and over guidance of tkinter(python) a complex structure of database can be created.

INTRODUCTION

This product is created to keep up records of offers, buy and staff records. The book shop gives the customers software-based shopping through an user interface. The motivation behind this product is to deal with the books in the bookshop. It includes the request handling and stock availability. We built up this product to keep up records of offers, buy and staff records.

In this system all records are saved in the database for generating a report i.e. Storing the information in a systematic way for future use. In present system during issuing order for more stock, the product table is required to check to availability of stock in hand, and it takes time to check records. In each process whether it is product management, maintaining customer transactions, payment management or making report, user has to pay attention to a greater extent while performing the tasks. Bookstore management software is for monitoring and controlling the transactions in a bookstore. Our software is easy to use for everyone, for administrators as well as beginners. It features a familiar and friendly, an attractive user interface, combined with strong searching insertion and selection capabilities.

BACKGROUND STUDY

In existing bookstores, all the transaction of books is done manually, so it takes a lot more time for a transaction like updating the number of books left after a book is bought, or searching for a member of books. Some of the problems being faced in the existing manual system are as follows:

1. Fast information retrieval is difficult.
2. Finding a book is tedious.
3. Information about issued books are not properly maintained.
4. No central database can be formed as information is not present in database.

The success of the interface depends largely on how clearly the problem is defined, carefully investigated and properly carried out through the choice of solution. During analysis, data is collected from various files. Records and transaction are handled by the new system. A good analysis model should provide not only the mechanisms of problem understanding but also the basic frame work of the solution.

The papers,

1. <https://prezi.com/ikbjuoirq6qh/book-store-management-system/>
2. “Research on Bookshop Management “ by Ms.Sneha Sakharkar, and Ms. Shubangi Karnuke

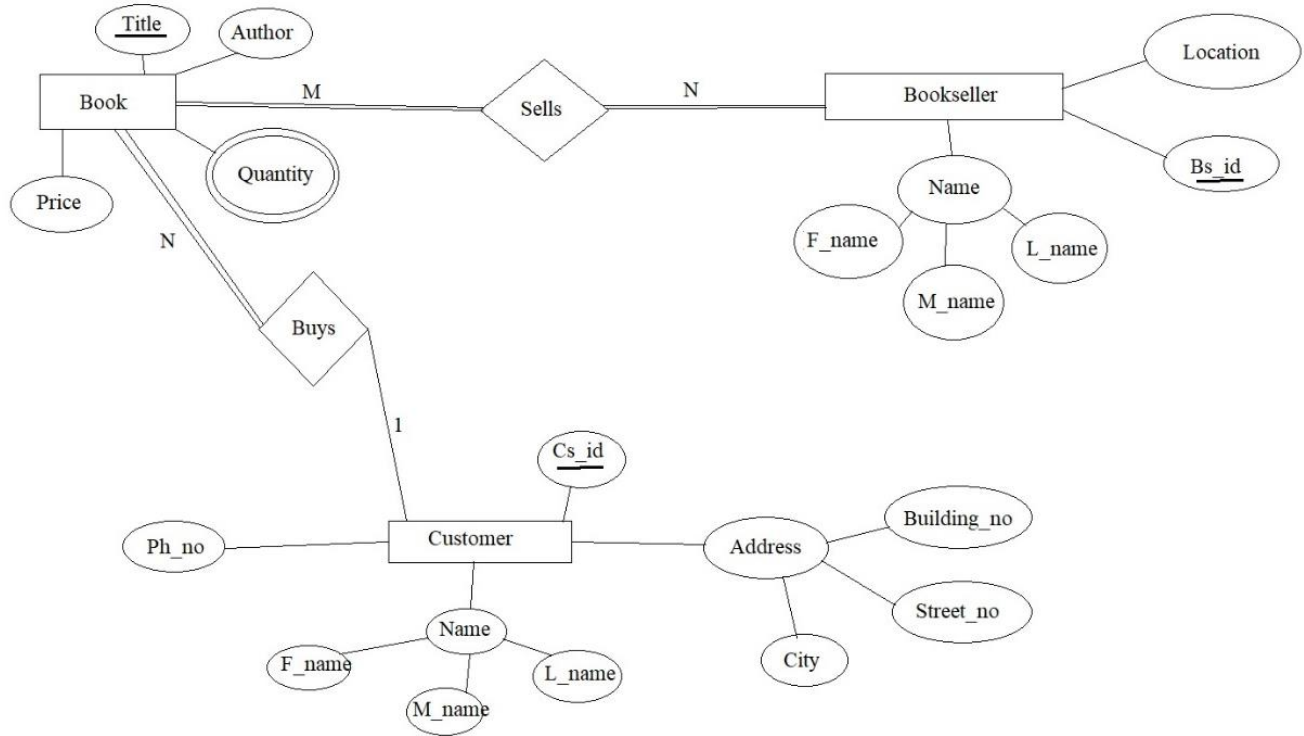
has helped us to understand the problems in the current bookstore system.

Studying these problems, we get the basic concept of our project. Majority of our study is from web materials and tutorials which helped us get a better understanding of tkinter and sqlite..

1. https://www.youtube.com/watch?v=RJB1Ek2Ko_Y&list=PL6gx4Cwl9DGBwibXFtPtflztSNPGuIB_d
2. <https://www.javatpoint.com/python-tkinter>
3. <https://www.sqlitetutorial.net/>
4. <https://www.youtube.com/watch?v=YIP1ORdBdKk&list=PL2Zn0eWTCPhTo2w5gQ9FqWcKPZDMqHM2O>

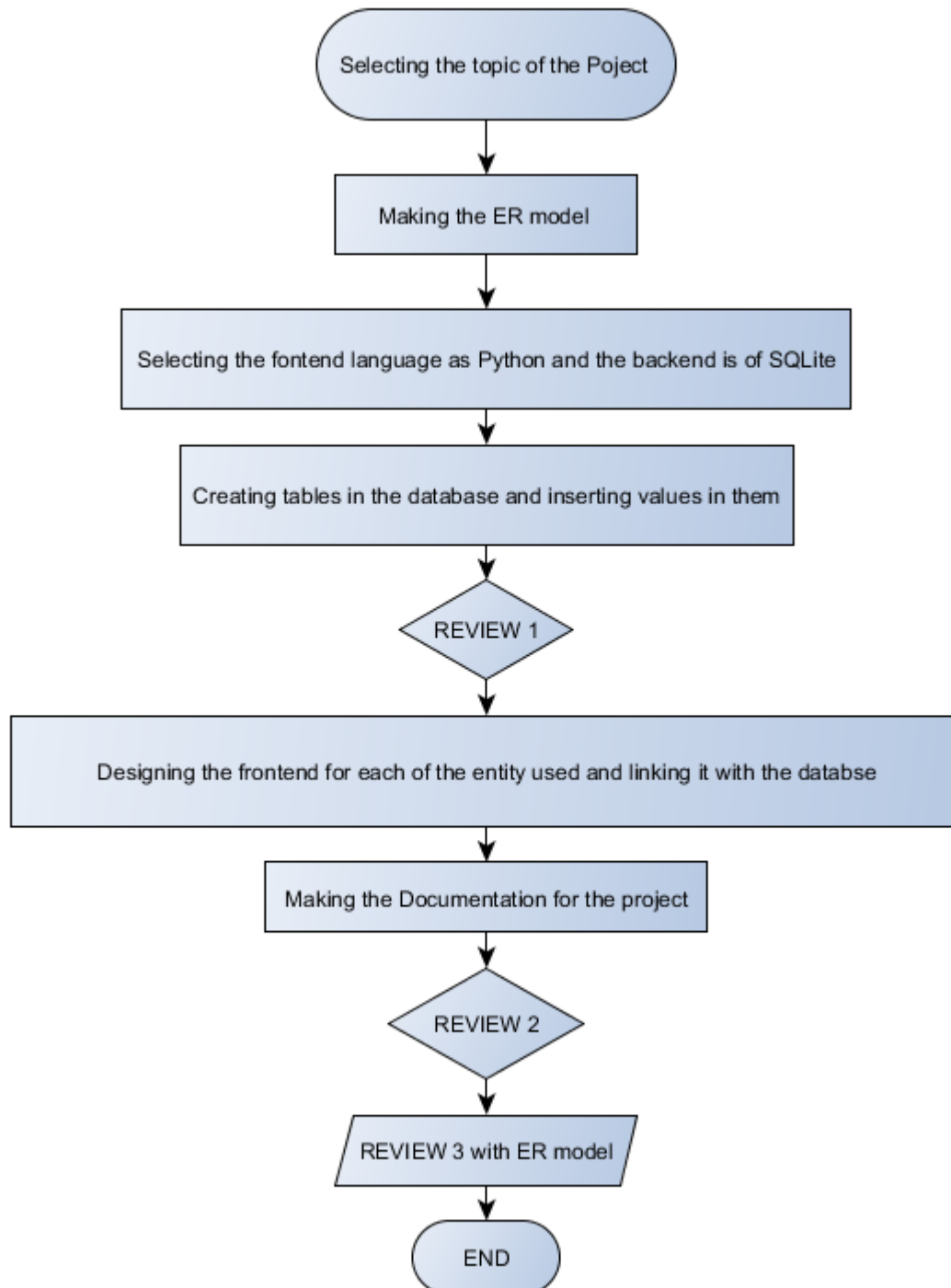
These links have helped us get a better understanding in building the graphic user interface and connecting it with the database.

ER DIAGRAM



METHODOLOGY

WORKFLOW



MODULES

The basic framework, the graphic user interface or the frontend of the project is created by python(tkinter). We create the window, labels, buttons, insert picture in the background and list showing the data we want in the respective windows which we use to store the information about the bookstore, customer and bookseller. We also create a function to help us link the data with the backend of the project.

The backend is programmed with sqlite3, where we create separate databases for bookstore, customer and employee to store the records. We create separate queries which help us to retrieve the data we want using 'select'. The database is connected to the frontend using 'connect'. Both frontend and backend help us do the following functions: -

- 1) Registration:** In the Registration module, all user can register their name, address, and mobile number, email id for contact. All the customers and bookseller data are stored in the database, Registration is the important process as information of the customer and bookseller are stored which may be needed later if any problem arises.
- 2) Search the Book:** In this module, user as well as bookstore manager can search any books by using name of book, name of author. It is a new search tab that lets our users perform simple searches of book content on our interface. User have the option of searching any books which are available to buy.
- 3) Purchase the Book:** In this module, user can purchase any kinds of books from our bookstore without any complexity. It is upto the seller, indicating relevant information about what they want to purchase, the quality, the price for that particular product or services.
- 4) Update or delete information:** In this module, the bookstore manager can update or delete information about books which are damaged or need some special care or are replaced.
- 5) Finding number of books:** This module takes care of the total number of books left in the bookstore. This number is updated if customer buys books from the store or some new books are added in the store. We can also find the which book has the highest or lowest cost for the convenience of the customers.

RESULTS AND DISCUSSION

SAMPLE CODE

FRONTEND USING PYTHON

```
from tkinter import *
from tkinter import messagebox
from PIL import Image,ImageTk
import backend2
def exitt():
    exit()

def bookstore():
    def customers():
        import backend3
        def get_selected_row1(event):
            global selected_tuple
            index=list2.curselection()[0]
            selected_tuple=list2.get(index)
            e1.delete(0,END)
            e1.insert(END,selected_tuple[1])
            e2.delete(0,END)
            e2.insert(END,selected_tuple[2])
            e3.delete(0,END)
            e3.insert(END,selected_tuple[3])
            e4.delete(0,END)
            e4.insert(END,selected_tuple[4])
            e5.delete(0,END)
            e5.insert(END,selected_tuple[5])
            e6.delete(0,END)
            e6.insert(END,selected_tuple[6])
```

```

e7.delete(0,END)
e7.insert(END,selected_tuple[7])
e8.delete(0,END)
e8.insert(END,selected_tuple[8])

```

```

def view_command1():
    list2.delete(0,END)
    for row in backend3.view():
        list2.insert(END,row)

```

Similar functions are used for adding, searching and modifying....

```

root3=Tk()
root3.geometry('750x750')
root3.wm_title("Bookseller Information",)
root3.configure(background = 'AntiqueWhite1')
imag3=Image.open("bookseller.png")
imag3=imag3.resize((750,750),Image.ANTIALIAS)
photos=ImageTk.PhotoImage(imag3)
lab1=Label(image=photos)
lab1.place(x=0,y=0,relwidth=1,relheight=1)

```

These are used for creating windows used in the GUI.....

```

l3=Label(root3,text="Middle_Name",relief='sunken',font=("Times New Roman",12))
l3.place(x=80,y=150)
middle_name_text=StringVar()
e3=Entry(root3,textvariable=middle_name_text)
e3.place(x=180,y=153)

```

is used for creating labels in the window.....

```

list3=Listbox(root3,background='light yellow', height=20,width=50)
list3.place(x=400,y=380)
sb1=Scrollbar(root3)

```



```

sb1.place(x=704,y=500)

list3.configure(yscrollcommand=sb1.set)

sb1.configure(command=list3.yview)

list3.bind('<<ListboxSelect>>',get_selected_row3)

b1=Button(root3,text="View all", width=12,command=view_command3)

b1.place(x=500,y=90)

```

is used for creating list and buttons in GUI.....

BACKEND USING SQLITE

```

import sqlite3

def connect():

    conn=sqlite3.connect("bk2.db")

    cur=conn.cursor()

    cur.execute("CREATE TABLE IF NOT EXISTS book_p (id INTEGER PRIMARY KEY,
title text, author text, quantity INTEGER, price INTEGER)")

    conn.commit()

    conn.close()

def insert(title,author,quantity,price):

    conn=sqlite3.connect("bk2.db")

    cur=conn.cursor()

    cur.execute("INSERT INTO book_p VALUES
(NULL,?,?,?,?)",(title,author,quantity,price))

    conn.commit()

    conn.close()

    view()

def view():

    conn=sqlite3.connect("bk2.db")

    cur=conn.cursor()

    cur.execute("SELECT * FROM book_p")

```

```
rows=cur.fetchall()
```

```
conn.close()
```

```
return rows
```

```
def search(title="",author="",quantity="",price=""):
```

```
    conn=sqlite3.connect("bk2.db")
```

```
    cur=conn.cursor()
```

```
    cur.execute("SELECT * FROM book_p WHERE title=? OR author=? OR quantity=? OR  
price=?", (title,author,quantity,price))
```

```
    rows=cur.fetchall()
```

```
    conn.close()
```

```
    return rows
```

```
def delete(id):
```

```
    conn=sqlite3.connect("bk2.db")
```

```
    cur=conn.cursor()
```

```
    cur.execute("DELETE FROM book_p WHERE id=?", (id,))
```

```
    conn.commit()
```

```
    conn.close()
```

```
def update(id,title,author,quantity,price):
```

```
    conn=sqlite3.connect("bk2.db")
```

```
    cur=conn.cursor()
```

```
    cur.execute("UPDATE book_p SET title=?, author=?, quantity=?, price=? WHERE  
id=?", (title,author,quantity,price,id))
```

```
    conn.commit()
```

```
    conn.close()
```

```
def total():
```

```
    conn=sqlite3.connect("bk2.db")
```

```
    cur=conn.cursor()
```

```
cur.execute("SELECT sum(quantity) FROM book_p")
rows=cur.fetchall()
val=int(str(rows[0]).strip('(),'))
return val
conn.close()
```

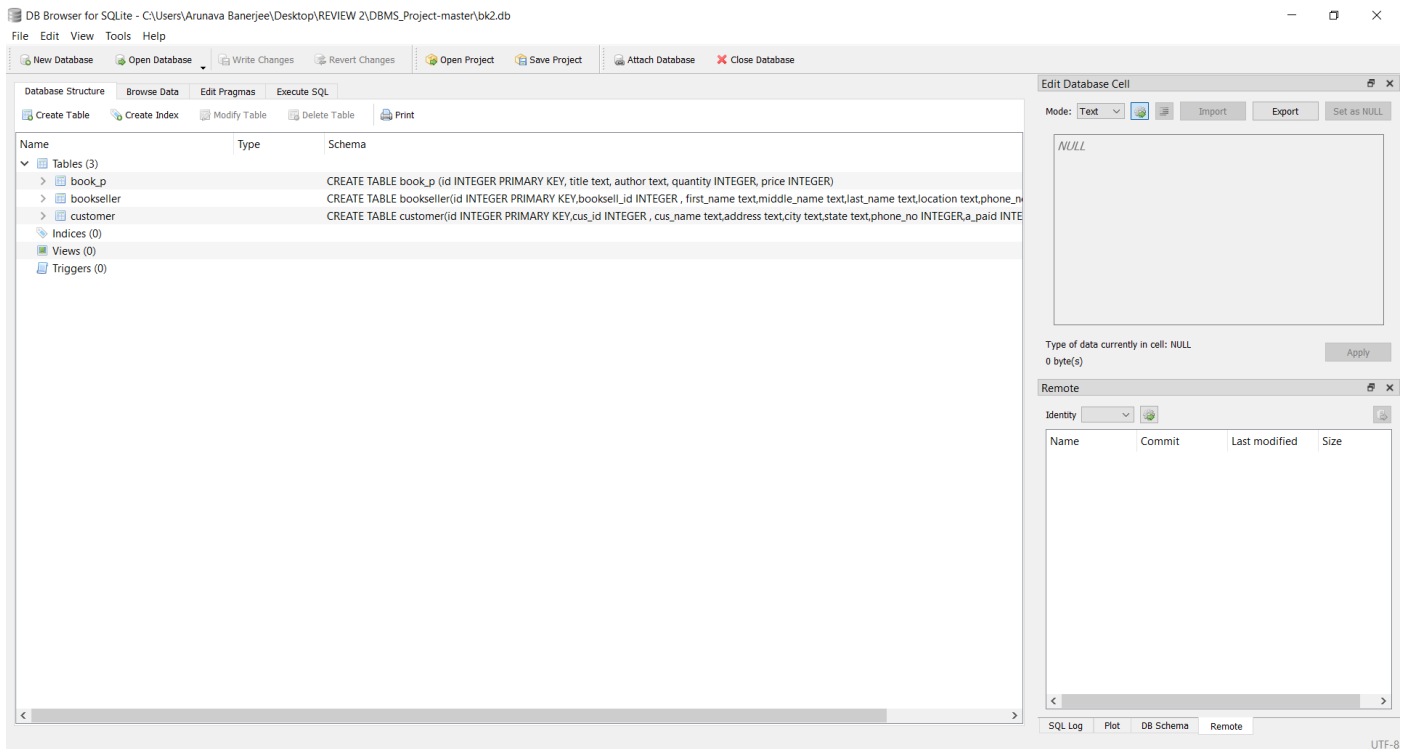
```
def maxval():
    conn=sqlite3.connect("bk2.db")
    cur=conn.cursor()
    cur.execute("SELECT max(price) FROM book_p")
    rows=cur.fetchall()
    val=int(str(rows[0]).strip('(),'))
    return val
    conn.close()
```

```
def minval():
    conn=sqlite3.connect("bk2.db")
    cur=conn.cursor()
    cur.execute("SELECT min(price) FROM book_p")
    rows=cur.fetchall()
    val=int(str(rows[0]).strip('(),'))
    return val
    conn.close()
```

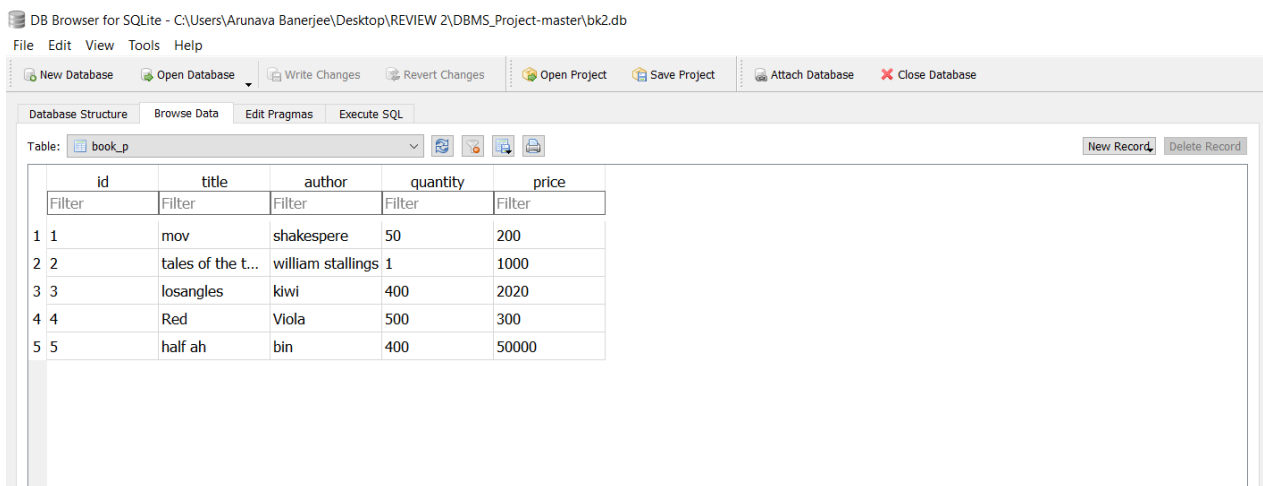
```
connect()
```

This is an example of how tables are created and various queries are used to perform operations in sqlite3. We have created 2 other tables for our project.

TABLES CREATED IN BACKEND

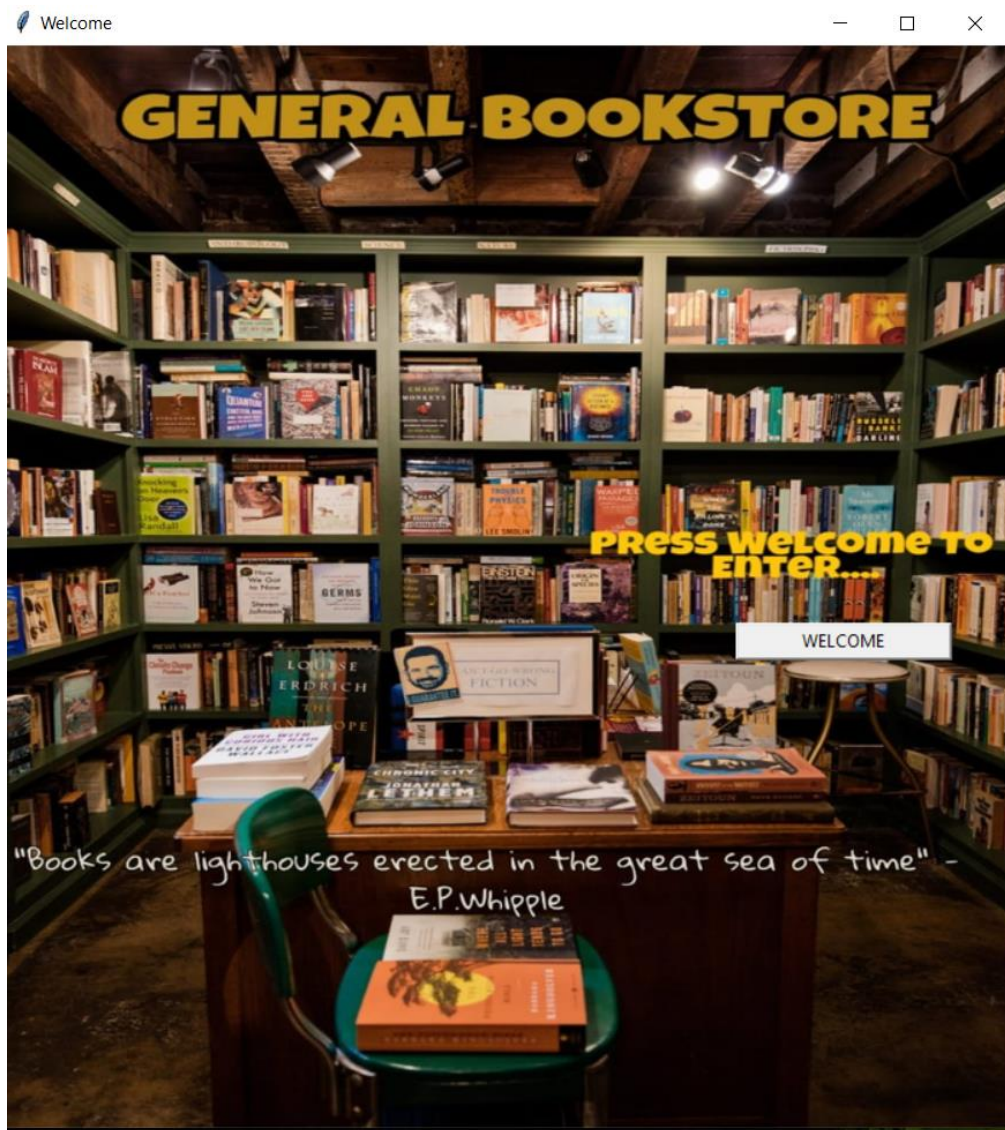


EG. TABLE BOOK_P

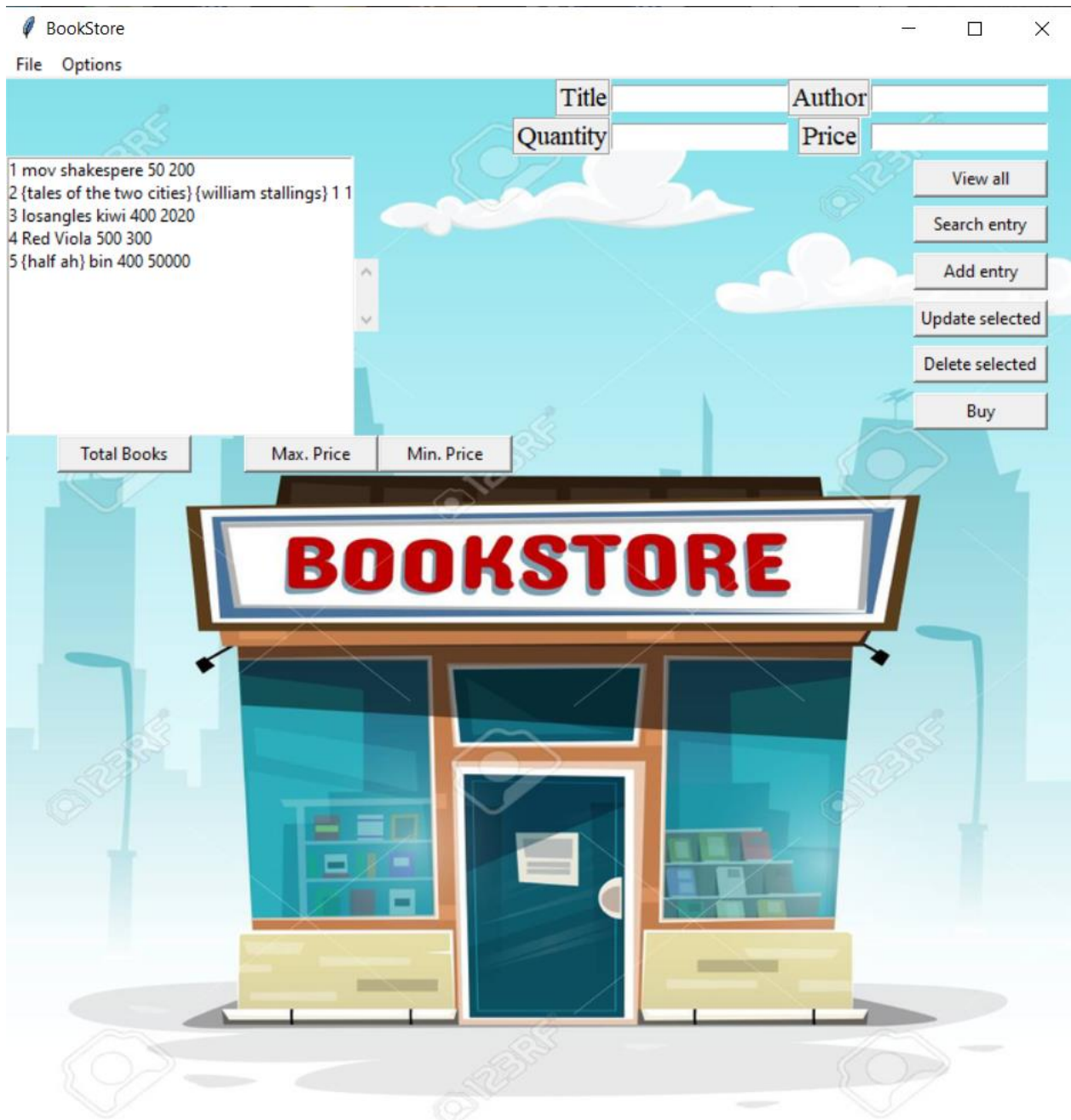


Modules in the GUI-

Opening page:



Book information



Customer Information:

Customer Information

BOOK STORE

Customer ID

Cus_Name

Address

City

State

Phone_no

Amount Paid

Total books

View all

Search entry

Add entry

Update selected

Delete selected

Total Books

Exit Bookstore

1 1234 rachit {q block} vellore {tamil nadu} 2569 2000 4

2 521 {Binamra } {q 214} vellore {tamil nadu} 9654238715 5

3 1230 binamra kathamandu lalit kath 56478 230 2

Bookseller Information:

Bookseller Information

Bookseller Information

Bookseller ID	<input type="text"/>	View all
First_Name	<input type="text"/>	Search entry
Middle_Name	<input type="text"/>	Add entry
Last_Name	<input type="text"/>	Update selected
Location	<input type="text"/>	Delete selected
Phone_no	<input type="text"/>	Total no.of Booksellers
Revenue	<input type="text"/>	Exit Bookstore

1 1234 Taylor Alison Swift London 9874568520 1000000
2 5678 will austin smith california 8748962581 50000

End Page :

Thanks for using .

— □ ×

"I Do Believe Something Very
Magical Can Happen When You
Read A Book." - J.K.Rowling

THANK YOU FOR USING THE
BOOKSTORE!!



EXAMPLE OF AN OPERATION

ADDING ENTRY TO BOOKSTORE

The bookseller information shown in the picture above has two instances. We can add instances from the GUI itself.

Instance to be added is entered in the following rows.

Bookseller Information

Bookseller Information

Bookseller ID	5678	View all
First_Name	Arunava	Search entry
Middle_Name	Kumar	Add entry
Last_Name	Banerjee	Update selected
Location	Kolkata	Delete selected
Phone_no	9674547276	Total no.of Booksellers
Revenue	6000	Exit Bookstore

1 1234 Taylor Alison Swift London 9874568520 1000000
2 5678 will austin smith califrnia 8748962581 50000

After adding,

Bookseller Information

Bookseller Information

Bookseller ID	5678	View all
First_Name	Arunava	Search entry
Middle_Name	Kumar	Add entry
Last_Name	Banerjee	Update selected
Location	Kolkata	Delete selected
Phone_no	9674547276	Total no.of Booksellers
Revenue	6000	Exit Bookstore

1 1234 Taylor Alison Swift London 9874568520 1000000

2 5678 will austin smith califnia 8748962581 50000

3 5678 Arunava Kumar Banerjee Kolkata 9674547276 6000

The instance is added in the list which states the instance is added in the database.

Link having our project:

https://drive.google.com/open?id=1Vv6i8iq3ympOEnynnJwDSXUug1C_uWFG

Frontend contains the codes related to designing while the different backends are used to create different tables. A database bk2 is created for the project.

CONCLUSION

Bookstore management system report in python is an attempt to overcome the present in efficient and time-consuming process of locating reserving and purchasing quality reading materials available in the shop. Through software book shop solution, it provides an easy way of searching reserving and purchasing of books. It is worth analysing and identifying the benefits as it would directly influence the productivity of the shop.

This software can be easily implemented under various environments. Any education institute, like schools and colleges, can make use of it by providing information about author, content of the available books in their library. Modifications can be easily done according to requirements and when necessary. It can be used in any type of book shop for managing all the sales and purchased activities and managing the data records related to the bookstore.

REFERENCES

- [1] Six Sigma n.d.: Applying Six Sigma to Software Implementation Projects Retrieved 22March 2007 from <http://software.isixsigma.com/library/content/> Australian Computer Society, 2003, ASC Code of Ethics. Retrieved March 15 2007, from <http://www.acs.org.au.htm>
- [2] Elmira, R. and Navathe, S. 2004. Enhanced Entity Relationship and UML. In Fundamentals of Database Systems, 3rd Edition
- [3] Ramakrishnan, R. and Gehrke, J. 2003. The Relational Model In Database Management Systems, 3rdEdition
- [4]https://www.youtube.com/watch?v=RJB1Ek2Ko_Y&list=PL6gx4Cwl9DGBwibXFtPtflztSNPGuIB_d
- [5] <https://prezi.com/ikbjuoirq6qh/book-store-management-system/>
- [6] “Research on Bookshop Management “ by Ms.Sneha Sakharkar,and Ms. Shubangi Karnuke
- [7] <https://www.javatpoint.com/python-tkinter>
- [8] <https://www.sqlitetutorial.net/>
- [9]<https://www.youtube.com/watch?v=YIP1ORdBdKk&list=PL2Zn0eWTCPhTo2w5gQ9FqWcKPZDMqHM2O>