SAVITRIBAI PHULE PUNE UNIVERSITY

A PRELIMINARY PROJECT REPORT ON

"GROCERY STORE DATABASE MANAGEMENT"

SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (TE COMPUTER ENGINEERING)

Academic Year: 2017-18

By:

1. VAIGAVI IYER (TECOC321)
2. POORNIMA JOSHI (TECOC326)
3. AISHWARYA KATKAR (TECOC330)

Under The Guidance of

Prof. Sushma Vispute

DEPARTMENT OF COMPUTER ENGINEERING,

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR 26, NIGDI, PRADHIKARAN

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that, the project entitled

“Grocery Shop Database Management”

is successfully carried out as a mini project successfully submitted by following students of “PCET's Pimpri Chinchwad College of Engineering, Nigdi, Pune-44**”.**

Under the guidance of Prof. S.Vispute

In the partial fulfillment of the requirements for the T.E (Computer Engineering)

1. **Vaigavi Iyer (TECOC321)**
2. **Poornima Joshi (TECOC326)**
3. **Aishwarya Katkar (TECOC330)**

Prof. S.R.Vispute

Project Guide

ABSTRACT

A Grocery Store permits a customer to submit item requirement and a bill will be calculated. Owner of the store will be able to analyze the stock availability, selling ratio of products, top selling product, minimum selling product, best buyer and imported stock.

The project has code (logic) and GUI in Python, were as the databases are written in mysql. The user interface has facility to interact with the customer to serve his needs and the retailer to analyze the sales, products, and more. The data is extracted from the databases for certain purposes in the project and also the data is entered in the databases. In this we have used three databases: Dealer, customer, store. The concepts of databases like cursor, count and different queries are implemented. The basic idea behind this project is to have analysis of a grocery shop, its sales and availability.

INDEX

Chapter 1: Introduction

Problem Statement

To design a grocery store which allows customer to buy items and allows store manager to analyze his stock availability and stock import as well as the best buyer and bestselling product.

Project Idea

To create a GUI based project for a grocery store which will allow customer to buy items and give its bill and allow the shopkeeper to check his stocks availability, best buyer, best city import, bestselling product.

Motivation

It is difficult for a shopkeeper to maintain different records manually and trace back them. In order to make it efficient we can develop a database connected project for the management of stock and provides billing facility. Also it helps the customers to have an efficient shopping.

Scope

The scope of the project is that it will be very beneficial for the shopkeepers to maintain a record of their sales which is now a days maintained on paper. This project can be used in all the stores, small scale as well as large scale to manage a large amount of data of sales, goods, best buyer, availability of products and more. Also this project can be extended to the next level to use it in all the stores and supermarkets in the future.

Literature Survey/ Requirement Analysis

The requirements were gathered and analyzed on the basis of requests of the customers to have an efficient shopping system and also importantly the requirements of the shopkeepers to have a digitalized way to maintain the records and moreover maintain large data. And frequently analyze and to keep a track of goods availability and other parameters.

Chapter 2: Project Design

H/W, S/W, resources, requirements & their detail explanation

H/W REQUIREMENTS: Computer (processor)

S/W REQUIREMENTS: MySQL,Python (Tkinter module)

MySQL:

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

PYTHON:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

TKINTER

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

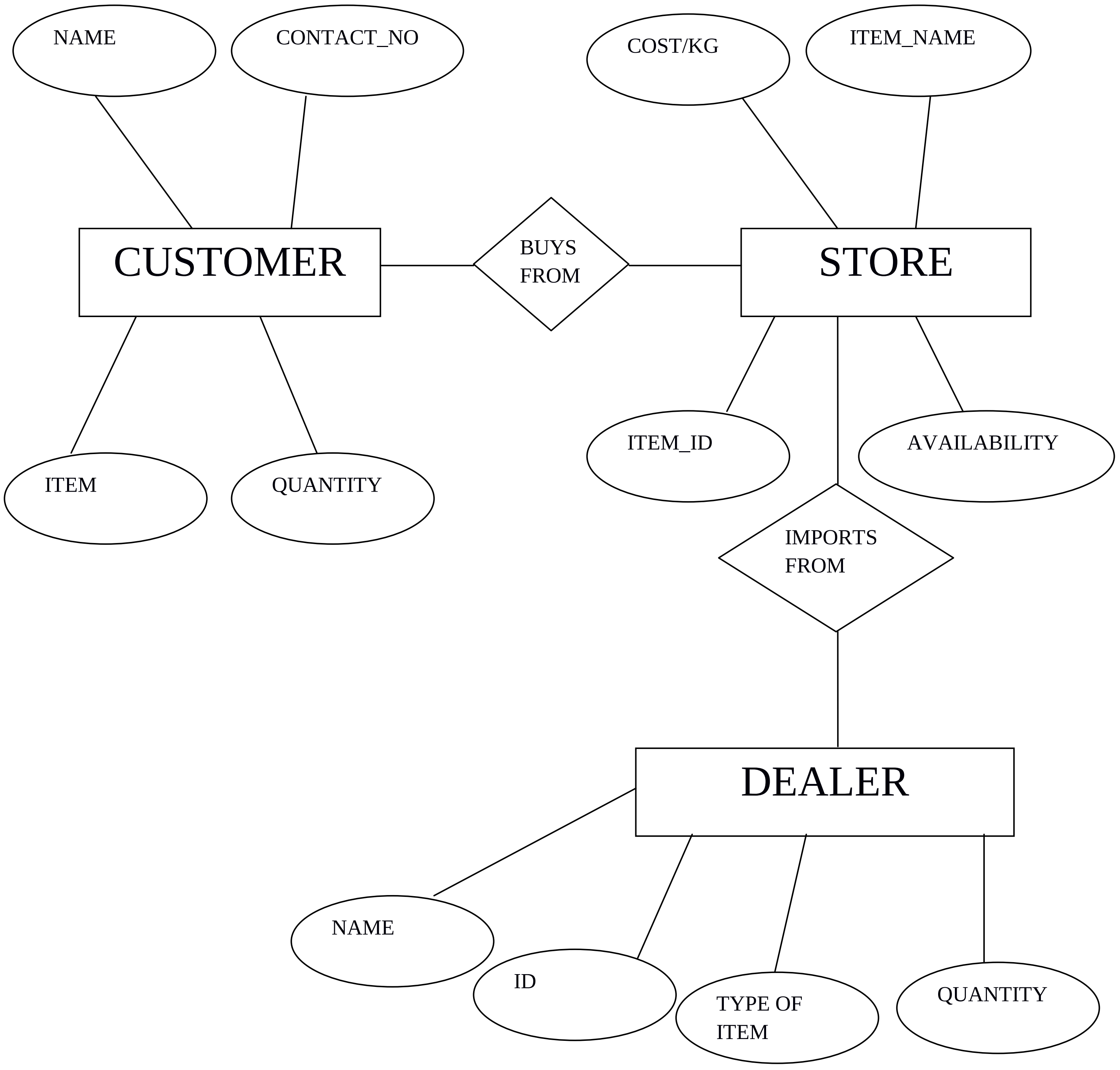
There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table −

Operator	Description
Button	The Button widget is used to display buttons in your application.
Canvas	The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
Checkbutton	The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at

	a time.
Entry	The Entry widget is used to display a single-line text field for accepting values from a user.
Frame	The Frame widget is used as a container widget to organize other widgets.
Label	The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
Listbox	The Listbox widget is used to provide a list of options to a user.
Menubutton	The Menubutton widget is used to display menus in your application.
Menu	The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
Message	The Message widget is used to display multiline text fields for accepting values from a user.
Radiobutton	The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
Scale	The Scale widget is used to provide a slider widget.
Scrollbar	The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
Text	The Text widget is used to display text in multiple lines.
Toplevel	The Toplevel widget is used to provide a separate window container.
Spinbox	The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of

	values.
PanedWindow	A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
LabelFrame	A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
tkMessageBox	This module is used to display message boxes in your applications.

E-R Model



NAME
CONTACT_NO
COST/KG
ITEM_NAME
CUSTOMER
BUYS FROM
STORE
ITEM
QUANTITY
ITEM_ID
AVAILABILITY
IMPORTS FROM
DEALER
NAME
ID
TYPE OF ITEM
QUANTITY

Schema of all the tables

CUSTOMER:

```
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| c_name  | varchar(17) | NO   |     | NULL    |       |
| mob_no  | int(11)     | NO   |     | NULL    |       |
| item    | varchar(12) | NO   |     | NULL    |       |
| quntity | int(11)     | NO   |     | NULL    |       |
| city    | int(11)     | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```

STORE:

```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| i_name       | varchar(12) | NO   |     | NULL    |       |
| i_id         | int(11)     | NO   |     | NULL    |       |
| availability | int(11)     | NO   |     | NULL    |       |
| costkg       | int(11)     | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

DEALER

```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| name     | varchar(26) | NO   |     | NULL    |       |
| d_id     | int(11)     | NO   |     | NULL    |       |
| type     | varchar(18) | NO   |     | NULL    |       |
| quantity | int(11)     | NO   |     | NULL    |       |
| city     | varchar(20) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

Chapter 3: Module Description

Block diagram with explanation of each module

SHOP
ITEM AVAILABILITY
BEST SELLING
BEST BUYER
CITYWISE IMPORT

1.SHOP:

This module enables customer to buy items and see the bill generated.

2.ITEM AVAILABILITY:

This modules enables shopkeeper to check and analyze the availability and storage in the shop.

3.BEST SELLING PRODUCT:

This modules enables shopkeeper to check and analyze the best selling product and improve its stock availability.

4.BEST BUYER:

This modules enables shopkeeper to check and analyze the best buyer   and provide them good discounting facilities.

5.CITYWISE INTAKE:

This modules enables shopkeeper to check and analyze the city wise import and maintain a record of best dealer.

Chapter 4: Results & Discussion

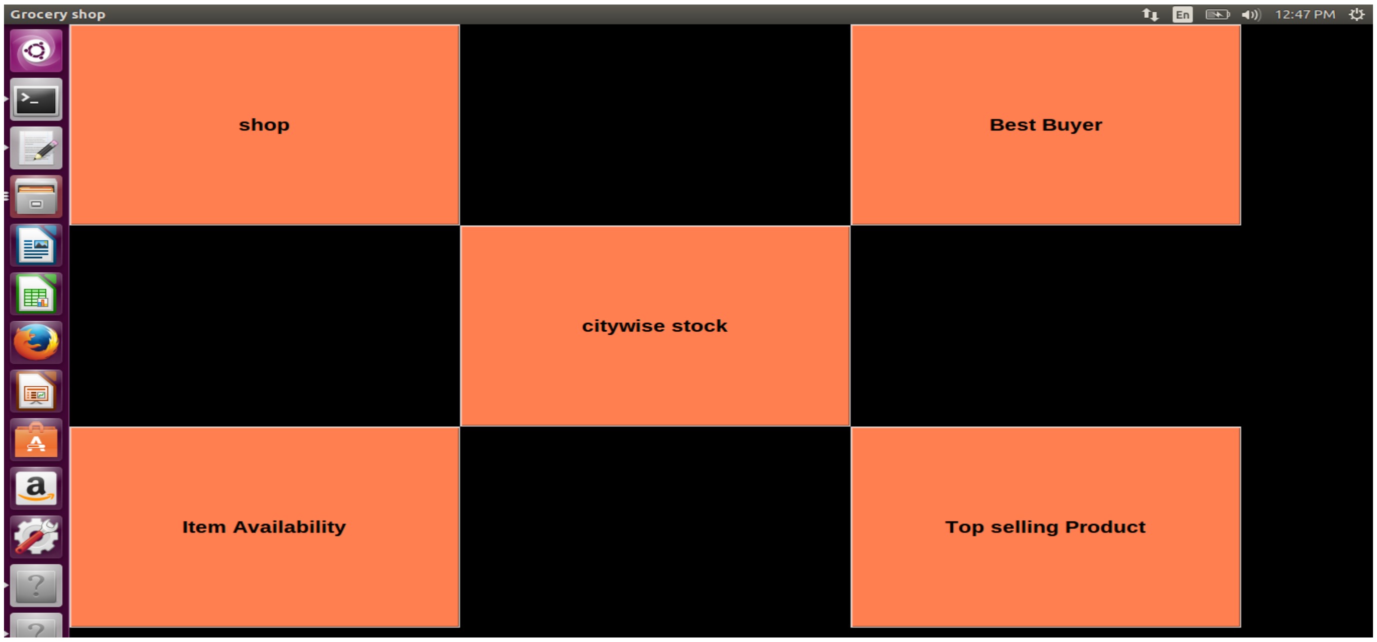
Source code

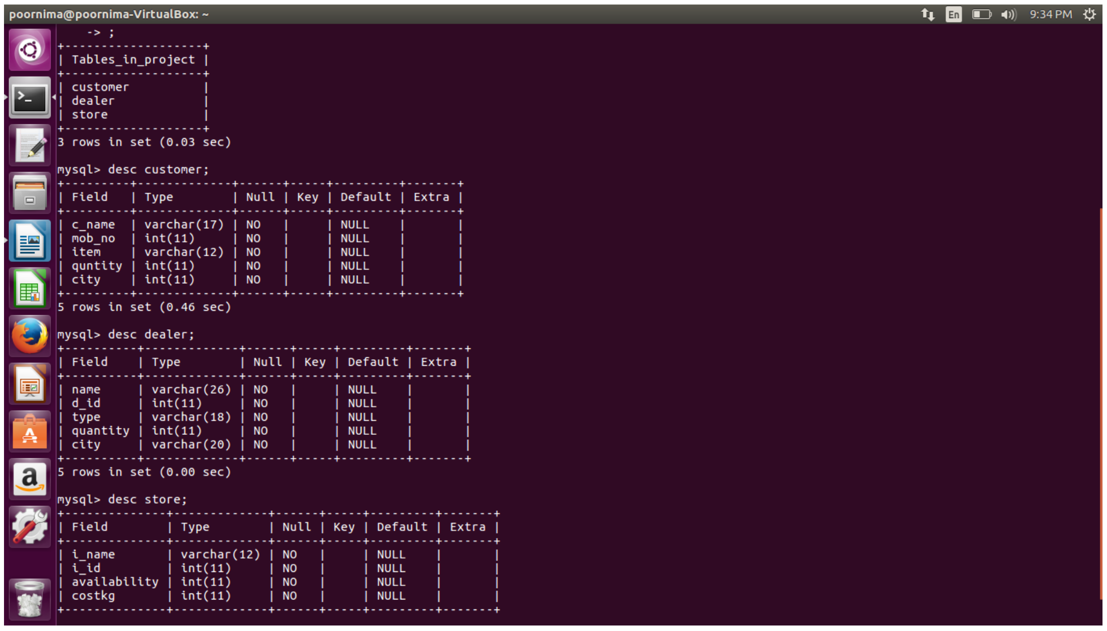
```
from Tkinter import *
#from Tkinter import messagebox
import MySQLdb
import numpy as np
import matplotlib.pyplot as plt
ax=plt.subplot(111)
db=MySQLdb.connect("localhost","root","root","project")
cursor=db.cursor()
colors=
    {1:'black',2:'blue',3:'brown',4:'green',5:'cyan',6:'magenta',7:'yellow',8:'red',9:'orang
e',10:'darkred',

    11:'hotpink',12:'indigo',13:'gold',14:'lavender',15:'khaki',16:'darkgreen',17:'darkred
',18:'darkorange',

    19:'darkblue',20:'crimson',21:'beige',22:'bisque',23:'azure',24:'aqua',25:'coral'}
root= Tk()

origin=0

def i_a():
    i=1
    sql1="SELECT availability from store"
    cursor.execute(sql1)
    y=cursor.fetchall()
    for row in y:
            n=colors[i]
            ax.bar(origin+i,row[0],width=0.5,color=n,align='center')
            i=i+1
            print(row)
    plt.xlabel('item name')
    plt.ylabel('availability')
    plt.title('item VS availability')
    plt.legend(('Apple
juice','eggs','milk','bread','vegetables','proteins','rice','oil','wheat','dishwash','orange
juice',
    'snacks','toothpaste','chocolates','icecream','peppersprey','handwash'),loc=9,ncol=5
)
    plt.show()

def b_b():
    i=1
    sql2="SELECT sum(city) from customer group by c_name"
    cursor.execute(sql2)
    y=cursor.fetchall()
    for row in y:
            m=colors[i]
            ax.bar(origin+i,row[0],width=0.5,color=m,align='center')
            i=i+1
            print row

    plt.xlabel('customer name')
    plt.ylabel('total bill')
    plt.title('customer Vs bill')
    plt.legend(('Harold Greer',
    'Dorothy Simon','Trevor Arnold','Bert Hawkins ','Ralph Hernandez','Hannah
Webster','Al Porter','Archie Joseph',
    'Amy Fitzgerald','Terence Nunez','Terry May','Carlos Taylor','Albert
Fuller','Connie Caldwell','Carroll Banks',
    'Karen Cain','Sheldon Sutton','Luis Rodriquez','Leonard Miller','Gwendolyn
Nelson','Lola Watson ','Gwen Martin ','Brenda          Becker','Pat
Simmons','poornima' ),loc='right')
    plt.show()



def t_s():
    i=1
    sql3="SELECT count(quntity) from customer group by item"
    cursor.execute(sql3)
    y=cursor.fetchall()
    for row in y:
            o=colors[i]
            ax.bar(origin+i,row[0],width=0.5,color=o,align='center')
            i=i+1
            print row
    plt.xlabel('item')
    plt.ylabel('QUANTITY')
    plt.title('item Vs quantity bought')
    plt.legend(( 'apple juice','eggs ','milk','bread','vegetables', 'protein','rice',
    'oil','wheat','dishwash','detergent','orange juice
    ','snack','toothpaste','chocoltes','icecream','pepperSpray','handwash'),loc='right')
    plt.show()

def c_s():
    i=1
    sql4="SELECT sum(quantity) from dealer group by city"
    cursor.execute(sql4)
    y=cursor.fetchall()
    for row in y:
            p=colors[i]
            ax.bar(origin+i,row[0],width=0.5,color=p,align='center')
            i=i+1
            print row

    plt.xlabel('city ')
    plt.ylabel('QUANTITY')
    plt.title('city Vs quantity bought')
    plt.legend((
'pune','banglore','sangli','mahabaleshwar','malkapur','kolhapur','ratnagiri','ujjain','kond
hwa',
    'konkan','goa','panjim','beed','shirdi',' hyderabad',' mumbai',' gadchiroli',' kolkata','
indapur'),loc=9,ncol=5)
    plt.show()
def buy():
    top=Toplevel()
    def get_():
            name=str(en.get())
            item=str(ei.get())
            mob=em.get()
            qt=int(eq.get())
            print name
            print item
            print mob
            print qt
            sql="SELECT costkg FROM store WHERE i_name= '%s'" %item
```

```
        cursor.execute(sql)
        y=cursor.fetchone()
        p=sum(y)
        print "p is:",p
        cst=qt*p
        print cst
        #print "final cost is:",
        eb=Label(top,text='%d'%(cst)).grid(row=5,column=1)
l1=Label(top,text='NAME:',relief=RAISED)
en=Entry(top)
l2=Label(top,text='ITEM NAME:',relief=RAISED)
ei=Entry(top)
l3=Label(top,text='MOBILE NUMBER:',relief=RAISED)
em=Entry(top)
l4=Label(top,text='QUANTITY:',relief=RAISED)
eq=Entry(top)
b1=Button(top,text='SUBMIT',command=get_)
l5=Label(top,text='Total Amount:',relief=RAISED)
#eb=Entry(root,textvariable=cst)
#eb.grid(row=5,column=1)


l1.grid(row=0,column=0)
en.grid(row=0,column=1)

l2.grid(row=1,column=0)
ei.grid(row=1,column=1)

l3.grid(row=2,column=0)
em.grid(row=2,column=1)

l4.grid(row=3,column=0)
eq.grid(row=3,column=1)

b1.grid(row=4,column=1)

l5.grid(row=5,column=0)
#eb.grid(row=5,column=1)

#root.mainloop()
```

```
b1=Button(root,text ="item availability", command=i_a,activebackground='grey')
b2=Button(root,text ="Best Buyer", command=b_b,activebackground='grey')
b3=Button(root,text ="Top selling Product", command=t_s,activebackground='grey')
b4=Button(root,text ="citywise stock", command=c_s,activebackground='grey')
b5=Button(root,text="shop", command=buy,activebackground='grey')

b5.grid(row=0,column=0)
b1.grid(row=1,column=0)
b2.grid(row=2,column=0)
b3.grid(row=3,column=0)
b4.grid(row=4,column=0)
"""root= Tkinter.Tk()"""
#root.geometry("500X700")
root.geometry('500x700')
root.mainloop()
```

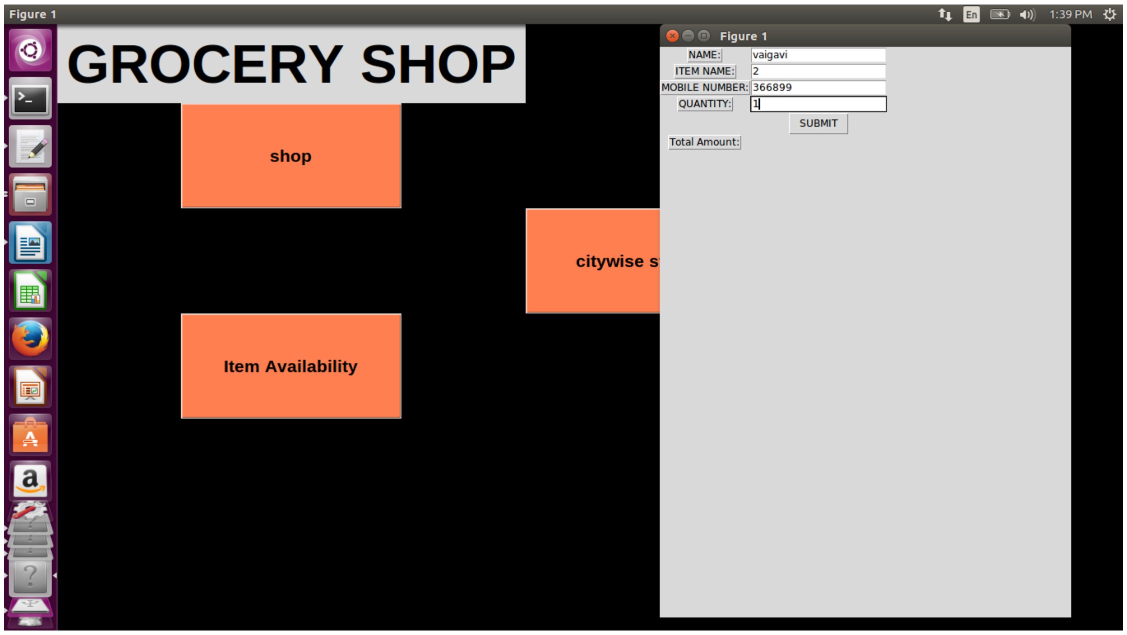
Screen shots including GUI:



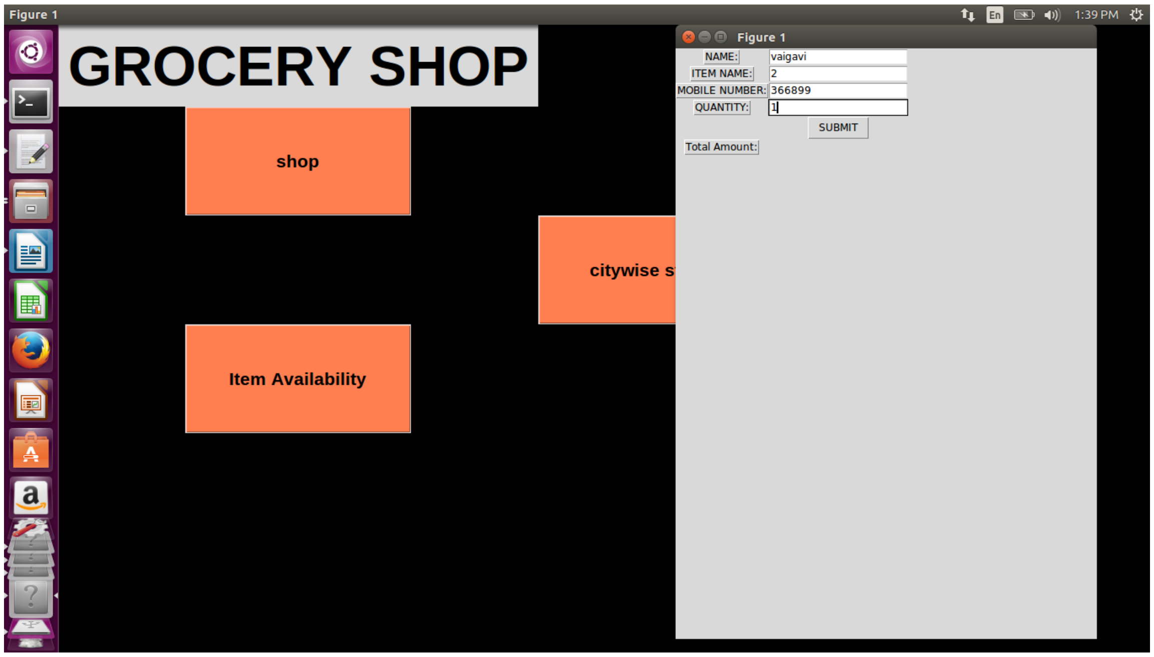


```
    -> ;
+-------------------+
| Tables_in_project |
+-------------------+
| customer          |
| dealer            |
| store             |
+-------------------+
3 rows in set (0.03 sec)

mysql> desc customer;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| c_name  | varchar(17) | NO   |     | NULL    |       |
| mob_no  | int(11)     | NO   |     | NULL    |       |
| item    | varchar(12) | NO   |     | NULL    |       |
| quntity | int(11)     | NO   |     | NULL    |       |
| city    | int(11)     | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
5 rows in set (0.46 sec)

mysql> desc dealer;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| name     | varchar(26) | NO   |     | NULL    |       |
| d_id     | int(11)     | NO   |     | NULL    |       |
| type     | varchar(18) | NO   |     | NULL    |       |
| quantity | int(11)     | NO   |     | NULL    |       |
| city     | varchar(20) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> desc store;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| i_name       | varchar(12) | NO   |     | NULL    |       |
| i_id         | int(11)     | NO   |     | NULL    |       |
| availability | int(11)     | NO   |     | NULL    |       |
| costkg       | int(11)     | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

Test Cases:



NAME: vaigavi
ITEM NAME: 2
MOBILE NUMBER: 366899
QUANTITY: 1



NAME: vaigavi
ITEM NAME: 2
MOBILE NUMBER: 366899
QUANTITY: 1

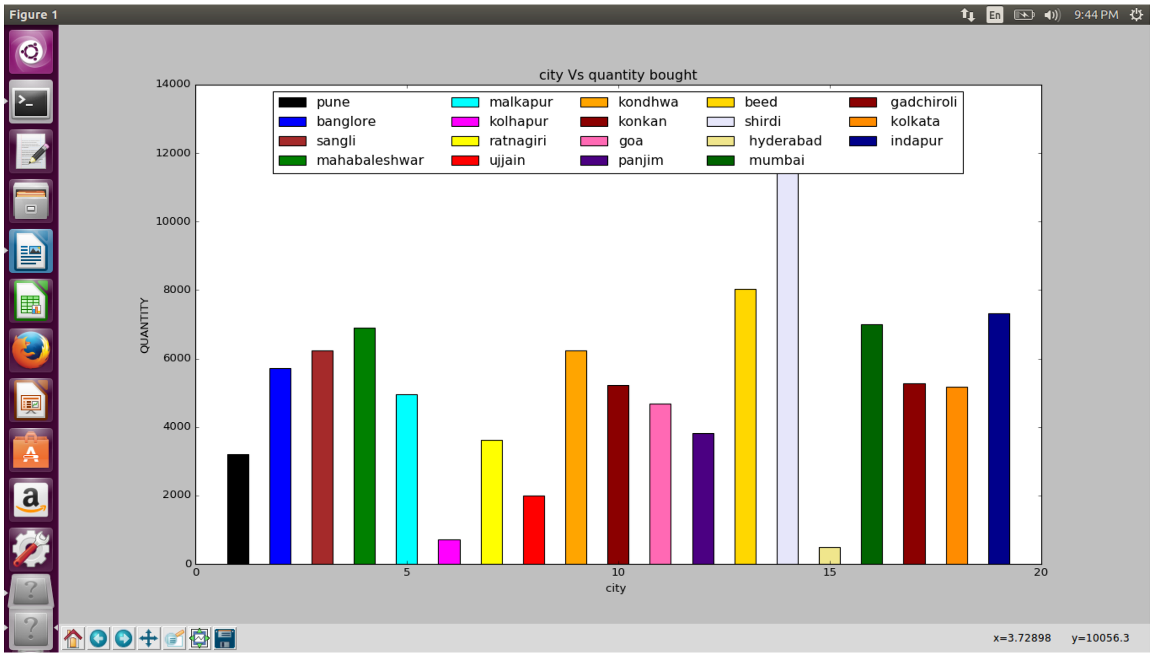


Figure 1
9:44 PM
city Vs quantity bought
pune
banglore
sangli
mahabaleshwar
malkapur
kolhapur
ratnagiri
ujjain
kondhwa
konkan
goa
panjim
beed
shirdi
hyderabad
mumbai
gadchiroli
kolkata
indapur
QUANTITY
city
x=3.72898 y=10056.3

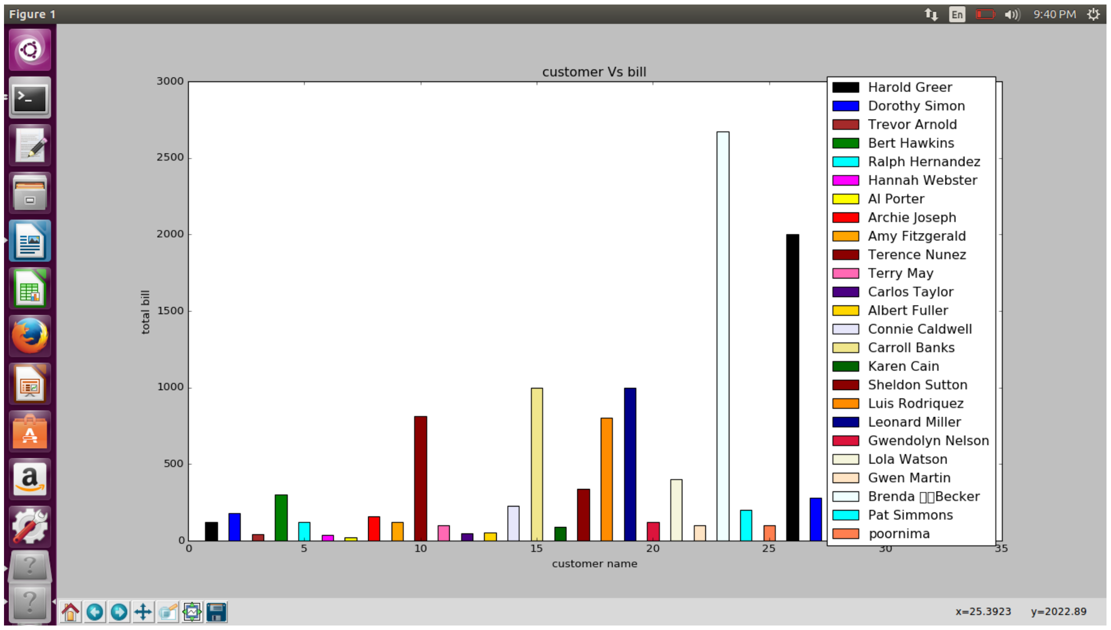


Figure 1
9:40 PM
customer Vs bill
Harold Greer
Dorothy Simon
Trevor Arnold
Bert Hawkins
Ralph Hernandez
Hannah Webster
Al Porter
Archie Joseph
Amy Fitzgerald
Terence Nunez
Terry May
Carlos Taylor
Albert Fuller
Connie Caldwell
Carroll Banks
Karen Cain
Sheldon Sutton
Luis Rodriquez
Leonard Miller
Gwendolyn Nelson
Lola Watson
Gwen Martin
Brenda Becker
Pat Simmons
poornima
total bill
customer name
x=25.3923 y=2022.89

Chapter 5: Conclusion

.

The main purpose involved in the making of this project was to provide efficient data handling and also provide value added services in the form of various analysis reports prepared by considering various parameters which will help in taking decisions useful for improvising the sales and services.

We have also provided a user friendly interface.

Eventually we have achieved the successful implementation of this project considering all the above parameters.