GoConstruct - Construction Management

Submitted in partial fulfilment for the requirements of ***EPJ component*** *of*

CSE2003
Data Structures and Algorithms

REVIEW 2 and 3

by

Siddharth Agrawal 19BCE2232

Pranav Tripathi 19BCE2240

Richard Daniel 19BCE0400

Nabbit Mahajan 19BCE0183

Under the guidance of

Prof. Kuruva Lakshmanna

School of Computer Science and Engineering

VIT, Vellore

Abstract

‘GoConstruct’ - Construction Tracking and Management software. We thought of this idea as the construction arena is some place where you usually don’t see a lot of technology involved in terms of facilitation and management. The customers need to visit the sites every once in a while to check the progress which can be done easily with the use of technology. Why roam around the markets when you can get the entire catalogue available with the help of databases.
We involve various essential entities in our project which match the real world scenario and try to make our project very functional in the real world.
GoConstruct is a construction management and tracking software built using Python and MySQL. Python was our choice of development as it has seamless integration with MySQL and we have utilised the Tkinter library of the Python language to form our UI components. With this project we wanted to showcase how DBMS can be applied to crude working environments. One doesn’t often see a lot of software involved in construction and hence our aim is to make the communication and management between the vendors and the customers really simple and functional.

Introduction

Background

Construction is an arena where management softwares are still untapped. Building a new home is a tedious task involving regular site visits and ensuring every small detail is executed. It becomes taxing for the vendor too as ensuring quality and deliver is a hard thing to manage. Customers keep looking for the best materials at the best prices and waste a lot of time in the process. Communicating with the vendors and placing orders requires time and physical presence which is dangerous in this time of COVID-19. The current construction management scenario can really be improved.

Objective

The above extrapolated problems are attempted to be solved by our project - GoConstruct. It used Tkinter in the frontend and MySQL as a backend. This project provides the Customer to the following facilities :

- Purchase items
- Make payments
- Place orders
- Give projects

The Vendor gets the following facilities :

- Request for payments
- Sell items
- Complete Orders
- Take projects
- Complete projects

Motivation

With the booming population, real estate is on a new high. Customer and vendor satisfaction in the journey of building a new home is crucial. Our software attempts to make the communication and management of almost everything related to construction seamless and enjoyable. With our simple UI, the customer be it of any age, need not learn anything new and just simply use it.

Project Resource Requirements

Software

- MySQL and Python

SQL is designed to query and extract data from tables within a database.
Python is particularly well suited for structured(tabular) data which can be fetched using SQL.
MySQLDB is an interface for connecting to a MySQL database server from Python.
It implements the Python database API, built on top of the MySQL API.
Python has many in-built libraries which help in data fetching from different sources.

- Tkinter

To put it simply, Tkinter is a Python library which helps us build the easiest and fastest GUI applications
To implement it : 'import tkinter as tk'
This module has two things which are absolutely necessary :
A. For any custom window : window = tk.Tk()
B. To enclose all widgets in a window, mainloop().

- VSCode

VSCode is a simple and lightweight text editor created by Microsoft. It was completely used for the entire coding process of the entire project.

Additional plugin used :

- ms-python.python
- Mysql connector
- Github for version control

Hardware

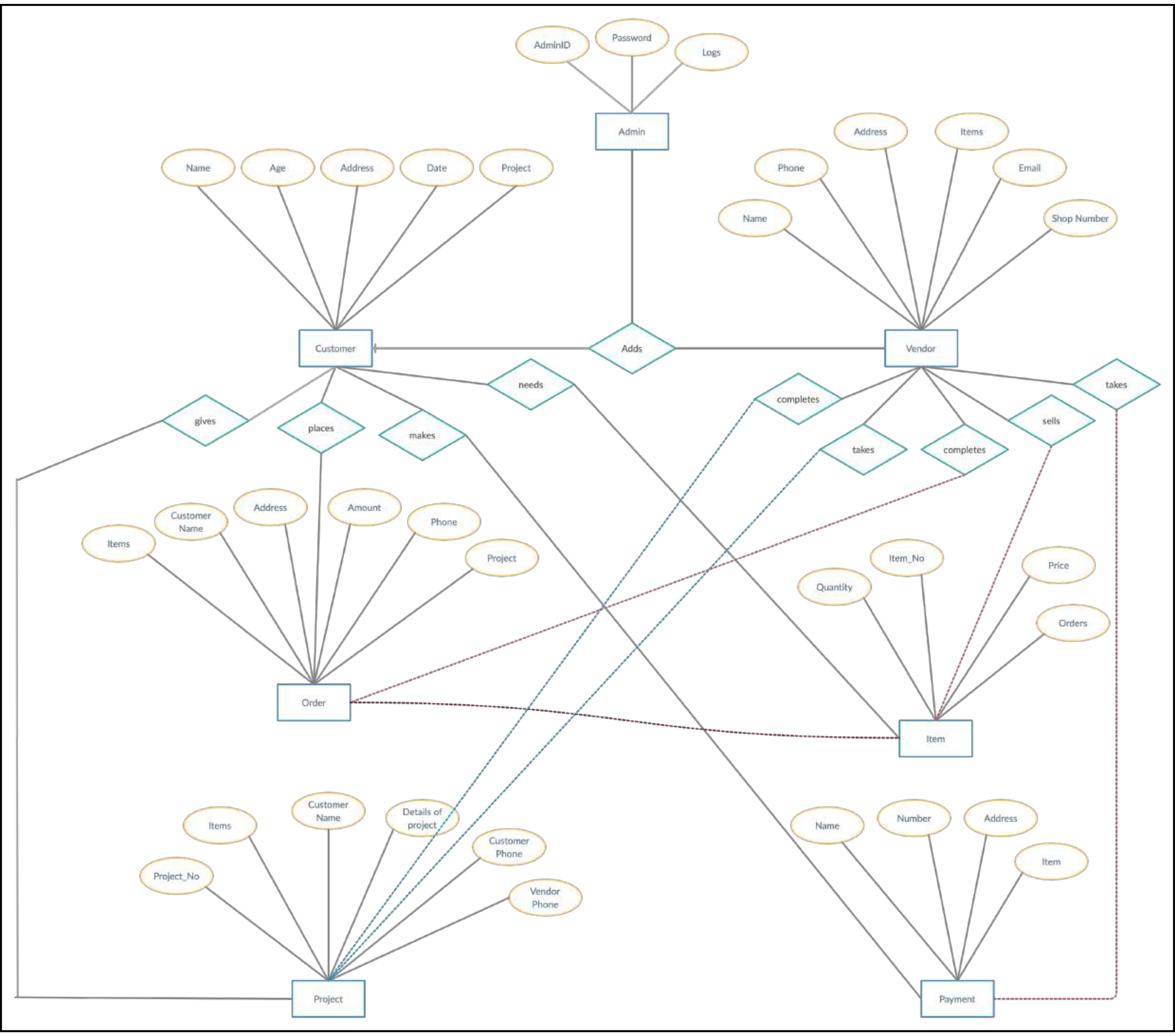
- Keyboard and Optical mouse for input
- Processor : Intel Pentium or higher
- Video display unit
- OS : MacOS 10 or higher, Ubuntu 18.04 or higher, Windows 7 or higher.

Literature Review

Serial Number	Research Paper (APA Format) and References
1	Suryadevara, N. K. GUI-based software development for sensor data collection, data extraction and data analysis using Python frameworks.
2	Lam, K. C., & Ng, S. T. (2006). A cooperative Internet-facilitated quality management environment for construction. *Automation in Construction*, *15*(1), 1-11.
3	Khaliq, K. A., Chughtai, O., Shahwani, A., Qayyum, A., & Pannek, J. (2019). An Emergency Response System: Construction, Validation, and Experiments for Disaster Management in a Vehicular Environment. *Sensors*, *19*(5), 1150.
4	Dynomant, E., Gorieu, M., Perrin, H., Denorme, M., Pichon, F., & Desfeux, A. (2017). MEDOC: a Python wrapper to load MEDLINE into a local MySQL database. *arXiv preprint arXiv:1710.06590*.
5	Jain, S. (2017). GeeksforGeeks: A computer science portal for geeks.(https://www.geeksforgeeks.org)
6	Kibert, C. J., & Hollister, K. C. (1994). An enhanced construction specific SQL. Automation in construction, 2(4), 303-312.
7	Musliman, I. A., Abdul-Rahman, A., & Coors, V. (2010). Incorporating 3D spatial operator with building information models in construction management using Geo-DBMS.
8	Chau, K. W., Cao, Y., Anson, M., & Zhang, J. (2003). Application of data warehouse and decision support system in construction management. Automation in construction, 12(2), 213-224.
9	plus2net(https://www.plus2net.com)
10	Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Design of the Project

ER diagram



NORMALIZATION OF TABLES

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

CUSTOMER

Date	**Name**	**Age**	**address**	**Project**
2020-09-12	Rahul	27	Street no. 5,Jaipur	Shopping Mall
2020-09-15	Harshit	26	MA Apartments,Kota	Hospital
2020-10-02	Raju	29	Gali no. 6 , New Delhi	Hotel

- CUSTOMER IS ALREADY IN 2NF FORM WE WILL CONVERT IT INTO 3NF FORM

NAME	AGE	ADDRESS
Rahul	27	Street no. 5,Jaipur
Harshit	26	MA Apartments,Kota
Raju	29	Gali no. 6 , New Delhi

ADMIN

Admin id	**Password**	**logs**
A101	101	
A102	102	
A103	103	

- ADMIN TABLE IS ALREDY IN 3NF FORM

VENDOR

Email	Name	Phone	Shop name	Address	Items
adityaconstructions@gmail.com	Aditya	9455844789	Aditya constructions	Near Aggrawal sweets , Noida	Cement
mohancon@gmail.com	Mohan	9877885542	Mohan constructions	Near marina beach,Chennai	Bricks
urcconstruction@gmail.com	Urvesh	8455698745	Urvesh constructions	Erode,Tamil Nadu	Paints

- VENDOR TABLE IS IN 1NF .WE WILL CONVERT IT INTO 2NF

EMAIL	NAME	ADDRESS
adityaconstructions@gmail.com	Aditya	Near Aggrawal sweets , Noida
mohancon@gmail.com	Mohan	Near marina beach,Chennai
urcconstruction@gmail.com	Urvesh	Erode,Tamil Nadu

AND

PHONE	SHOP NAME	ITEMS
9455844789	Aditya constructions	Cement
9877885542	Mohan constructions	Bricks
8455698745	Urvesh constructions	Paints

- NOW WE WILL CONVERT THIS INTO 3NF BY ADDING ONE MORE TABLE

SHOPNAME	NAME	ADDRESS
Aditya constructions	Aditya	Near Aggrawal sweets , Noida
Mohan constructions	Mohan	Near marina beach,Chennai
Urvesh constructions	Urvesh	Erode,Tamil Nadu

ITEM

QUANTITY	ITEM_NO	PRICE	ORDERS
30	I1001	60000	Cement
500	I2001	30000	Bricks
20	I3001	30000	Paint

- Items table is already in normalized 3NF form

ORDER

CUSTOMER_NAME	ITEMS	ADDRESS	AMOUNT	PHONE	PROJECT
Rahul	Bricks,Cement	Street no. 5,Jaipur	100000	9457885412	Shopping Mall
Harshit	Paint,Bricks	MA Apartments,Kota	90000	8888456523	Hospital
Raju	Paint,Cement	Gali no. 6 , New Delhi	100000	9788546523	Hotel

- Order table is already in 2Nf form .We will now convert this table into 3NF

CUSTOMER_NAME	ADDRESS
Rahul	Street no. 5,Jaipur
Harshit	MA Apartments,Kota
Raju	Gali no. 6 , New Delhi

And

ITEMS	AMOUNT	PHONE	PROJECT
Bricks,Cement	100000	9457885412	Shopping Mall
Paint,Bricks	90000	8888456523	Hospital
Paint,Cement	100000	9788546523	Hotel

PROJECT

Project no	Items	Cust_Name	Details	Cust_no	Vendor
P001	Bricks,Cement	Rahul	9457885412	C001	Aditya constructions
P002	Paint,Bricks	Harshit	8888456523	C002	Mohan constructions
P003	Paint,Cement	Raju	9788546523	C003	Urvesh constructions

- PROJECT TABLE IS IN 1NF FORM WE WILL CONVERT IT INTO 3NF FORM
- 4 TABLES WILL BE CREATED IN 3NF FORM

CUSTOMER_NO	CUST_NAME	DETAILS
C001	Rahul	9457885412
C002	Harshit	8888456523
C003	Raju	9788546523

PROJECT NO	ITEMS	VENDOR
P001	Bricks,Cement	Aditya constructions
P002	Paint,Bricks	Mohan constructions
P003	Paint,Cement	Urvesh constructions

CUST_NAME	DETAILS
Rahul	9457885412
Harshit	8888456523
Raju	9788546523

CUST_NO	CUST_NAME
C001	Rahul
C002	Harshit
C003	Raju

PAYMENT

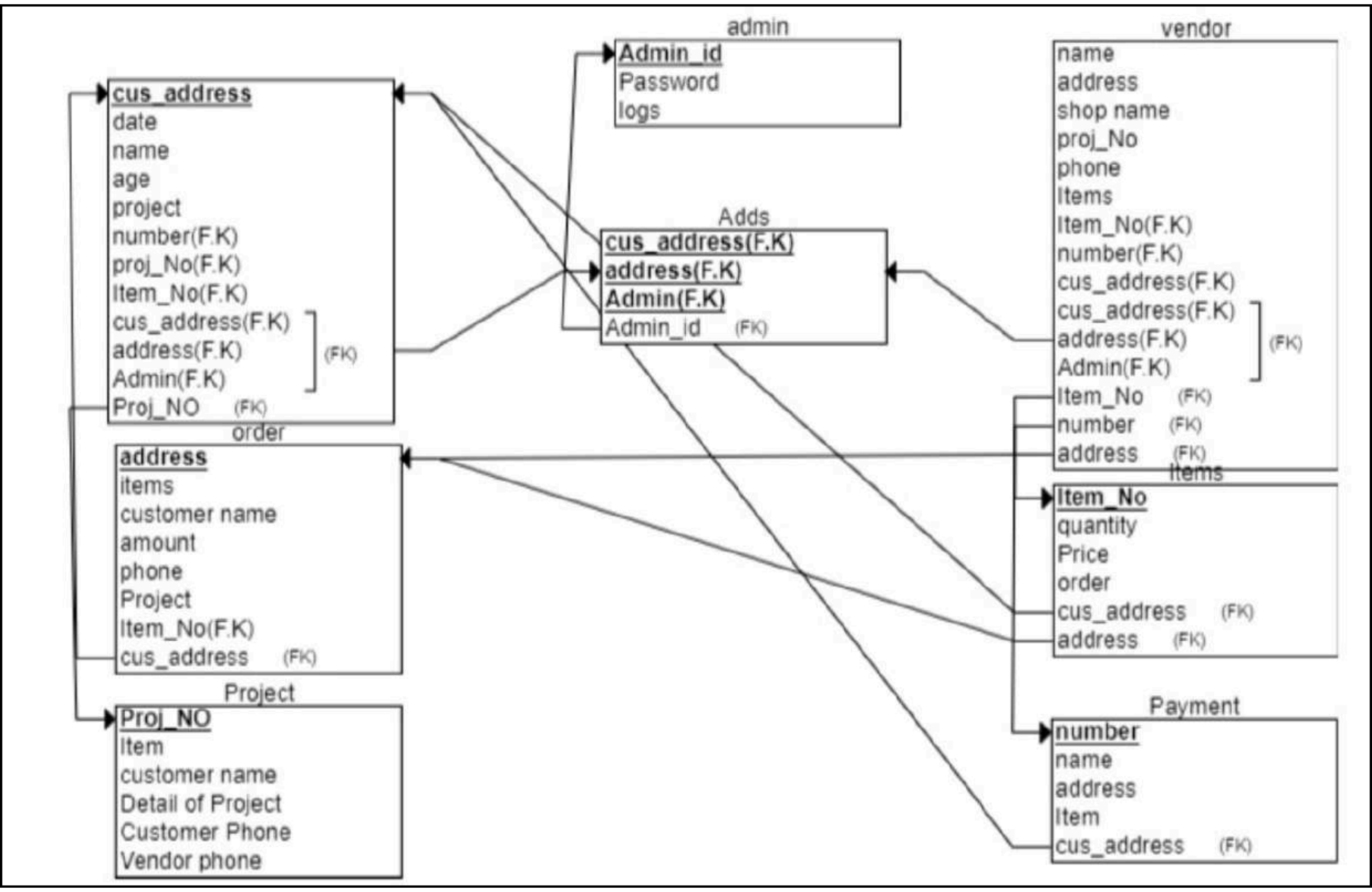
NAME	PHONE	ADDRESS	ITEMS
Rahul	9457885412	Street no. 5,Jaipur	Bricks,Cement
Harshit	8888456523	MA Apartments,Kota	Paint,Bricks
Raju	9788546523	Gali no. 6 , New Delhi	Paint,Cement

- PAYMENT IS IN 1NF WE WILL CONVERT IT INTO 3NF NORMALIZED FORM

NAME	ITEM	ADDRESS
Rahul	Bricks,Cement	Street no. 5,Jaipur
Harshit	Paint,Bricks	MA Apartments,Kota
Raju	Paint,Cement	Gali no. 6 , New Delhi

NAME	ADDRESS
Rahul	Street no. 5,Jaipur
Harshit	MA Apartments,Kota
Raju	Gali no. 6 , New Delhi

Relational Schema



Tables and Constraints

Customer

Field	Type	Null	Key	Default	Extra
Date	Date	No		NULL	
Name	Varchar	No		NULL	
Age	Int	No		NULL	
Address	Varchar	No	PRI	NULL	
Project	char	No		NULL	

Admin

Field	Type	Null	Key	Default	Extra
Admin_id	Int	No	PRI	NULL	
Password	varchar	No		NULL	
Logs	Char	No		NULL	

Vendor

Field	Type	Null	Key	Default	Extra
Email	Varchar	No	PRI	NULL	
Name	Varchar	No	PRI	NULL	
Phone	Bigint	No	PRI	NULL	
Shop_name	char	No		NULL	
Address	Varchar	No		NULL	
Items	Char	No		NULL	

Item

Field	Type	Null	Key	Default	Extra
Quantity	Int	No		NULL	
Item_no	Int	No	PRI	NULL	
Price	Int	No		NULL	
Orders	Char	No		NULL	

Order

Field	Type	Null	Key	Default	Extra
Cust_ame	Varchar	No		NULL	
Items	Int	No		NULL	
Address	Char	No		NULL	
Amount	Int	No		NULL	
Phone	Bigint	No	PRI	NULL	
Project	Char	No		NULL	

Project

Field	Type	Null	Key	Default	Extra
Project_no	Int	No	PRI	NULL	
Items	Int	No		NULL	
Cust_name	Varchar	No	PRI	NULL	
Details	Char	No		NULL	
Cust_no	Bigint	No	PRI	NULL	
Vend_no	Bigint	No		NULL	

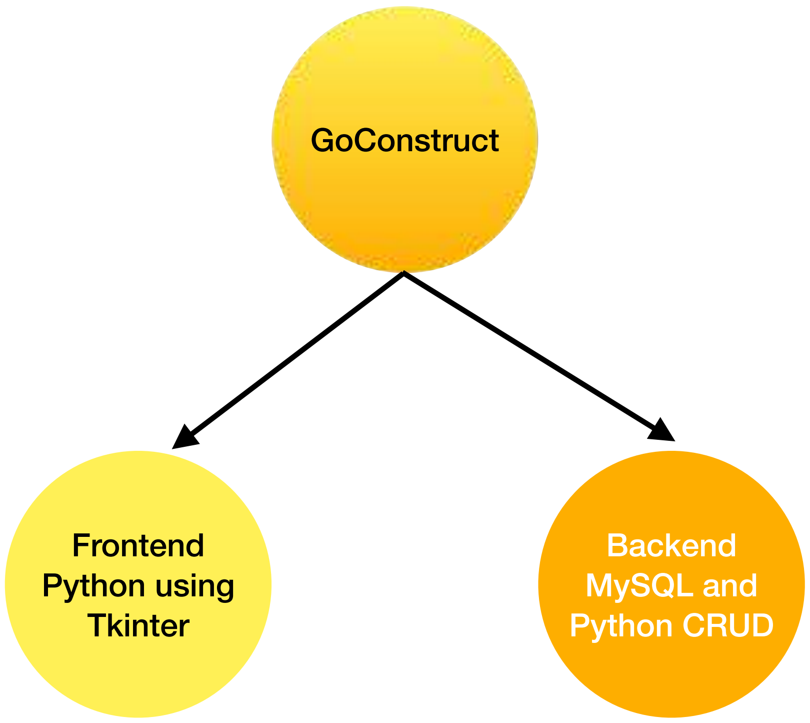
Payment

Field	Type	Null	Key	Default	Extra
Name	Varchar	No	PRI	NULL	
Phone	Bigint	No	PRI	NULL	
Address	Char	No		NULL	
Item	Char	No		NULL	

Implementation Overview

Tech stacks used :

Python **Tkinter** **MySQL**



Python was the language of choice for the frontend as the project creates a desktop application and Python's library Tkinter eases the creation of all the kinds of UI needed for this project. The mysql connector for Python provides with seamless connection to the database and also fast retrieval times. The cursor created with the help of the connector executes all the queries and also passes the data. The data is taken as input from the user using Tkinter TextBoxes and the get function. The data is then stored into variables and passed into the queries as arguments. The commit function of the mysql connector makes real changes to the connected database. Before running the software, the local mysql server needs to be started. The project has the following modules :

- Database creation
- Table creation
- Core code
    - Items : Needs{Consumer} and Sells{Vendor}
    - Payments : Makes{Consumer} and Takes{Vendor}
    - Orders : Places{Consumer} and Completes{Vendor}
    - Projects : Gives{Consumer}, Takes{Vendor} and Completes{Vendor}

Modules with codes

Database Creation

```
import mysql.connector

mydb = mysql.connector.connect(
  host="127.0.0.1",
  user="root",
  password="Chiku$!d9"
)

mycursor = mydb.cursor()

mycursor.execute("CREATE DATABASE Goconstruct")
```

Table Creation

```
import mysql.connector

mydb = mysql.connector.connect(
    host = "127.0.0.1",
    user = "root",
    password = "Chiku$!d9",
    database = "Goconstruct"
)

mycursor = mydb.cursor()

execute = [
"CREATE TABLE Vendor(Name VARCHAR(50), Phone BIGINT , Email
VARCHAR(50), ShopName VARCHAR(50), Address VARCHAR(50), Items
VARCHAR(50), PRIMARY KEY(Name, Phone))",
"CREATE TABLE Project(ProjectNumber INT, CustomerName
VARCHAR(100), CustomerNumber BIGINT, VendorNumber BIGINT, Item
INT, Details VARCHAR(50), PRIMARY KEY(ProjectNumber, CustomerName,
CustomerNumber))",
"CREATE TABLE Payment(Name VARCHAR(50), Phone BIGINT, Address
VARCHAR(50), Item VARCHAR(50), PRIMARY KEY(Name, Phone))"
]

for i in execute :
    mycursor.execute(i)

mydb.commit()

'''
DONE :
#customer : Name VARCHAR(100), Age INT, Address VARCHAR(20)
PRIMARY KEY, Project VARCHAR(20), Date DATE
#admin : AdminID INT PRIMARY KEY, Password VARCHAR(30), Logs
VARCHAR(30)
#item : ItemNumber INT PRIMARY KEY, Quantity INT, Price INT, Order
VARCHAR(50)
#order : CustomerName VARCHAR(50), Phone BIGINT PRIMARY KEY,
Address VARCHAR(50), Amount INT, Project VARCHAR(50), Item INT)
#vendor : Name VARCHAR(50), Phone BIGINT , Email VARCHAR(50),
ShopName VARCHAR(50),
# Address VARCHAR(50), Items VARCHAR(50), PRIMARY KEY(Name, Phone)
#project : ProjectNumber INT, CustomerName VARCHAR(100),
CustomerNumber BIGINT,
#VendorNumber BIGINT, Item INT, Details VARCHAR(50), PRIMARY
KEY(ProjectNumber, CustomerName, CustomerNumber)
#payment : Name VARCHAR(50), Phone BIGINT, Address VARCHAR(50),
Item VARCHAR(50), PRIMARY KEY(Name, Phone)'''
```

Core Code

Items

1. Needs

```
def NeedItem() :
    global tItem, tQuantity, tOT

    myCursor.execute("SELECT * FROM Item")
    my_wo = tkinter.Tk()
    my_wo.title("Available Items")
    my_wo.geometry("250x250")
    i=0
    for Item in myCursor:
        for j in range(len(Item)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Item[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Buy Items")
    l0 = tkinter.Label(my_w,  text='Needs 
Items',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Item Number : ', 
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tItem = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tItem.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Quantity : ', 
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tQuantity = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tQuantity.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Order : ', width=10,anchor="c" 
)
    l3.grid(row=5,column=1)
    tOT = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tOT.grid(row=5,column=2)

    b1 = tkinter.Button(my_w,  text='Buy', width=10,
command=lambda: delete_data_item())
    b1.grid(row=7,column=2)

def delete_data_item() :
    my_name = tItem.get("1.0",END)
    query="DELETE FROM `Item` WHERE ItemNumber = %s"
    myCursor.execute(query,(my_name,))
    db_connection.commit()
    tItem.delete('1.0',END)
    tQuantity.delete('1.0',END)
    tOT.delete('1.0',END)
    print("Query executed")
```

2. Sells

```
def SellsItems() :
    global t1, t2, t3, t4

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Needs Items")
    l0 = tkinter.Label(my_w,  text='Needs
Items',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Item Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    t1 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t1.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Quantity : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    t2 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t2.grid(row=4,column=2)
    l3 = tkinter.Label(my_w,  text='Price : ', width=10,anchor="c"
)
    l3.grid(row=5,column=1)
    t3 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t3.grid(row=5,column=2)

    l4 = tkinter.Label(my_w,  text='Order : ', width=10,anchor="c"
)
    l4.grid(row=6,column=1)
    t4 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t4.grid(row=6,column=2)

    b1 = tkinter.Button(my_w,  text='Put the item on sale',
width=10, command=lambda: add_data_item())
    b1.grid(row=7,column=2)

def add_data_item() :
    my_name = t1.get("1.0",END)
    my_class = t2.get("1.0",END)
    my_mark = t3.get("1.0",END)
    my_gender = t4.get("1.0",END)
    query="INSERT INTO  `Item`
(`ItemNumber` ,`Quantity` ,`Price` ,`OrderItem`) VALUES(%s,%s,%s,
%s)"
    my_data=(my_name,my_class,my_mark,my_gender)
    myCursor.execute(query,my_data)
    db_connection.commit()
    t1.delete('1.0',END)
    t2.delete('1.0',END)
    t3.delete('1.0',END)
    t4.delete('1.0',END)
    print("Query executed")
```

Payments

1. Makes

```
def MakePayment() :
    global tNamePay, tPhonePay, tAddressPay, tItemPay
    myCursor.execute("SELECT * FROM Payment")
    my_wo = tkinter.Tk()
    my_wo.title("Requested payments ")
    my_wo.geometry("250x250")
    i=0
    for Item in myCursor:
        for j in range(len(Item)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Item[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Pay")
    l0 = tkinter.Label(my_w,  text='Make
Payment',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Name : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tNamePay = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tNamePay.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Phone : ', width=10,anchor="c"
)
    l2.grid(row=4,column=1)
    tPhonePay = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tPhonePay.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Address : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    tAddressPay = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tAddressPay.grid(row=5,column=2)

    l3 = tkinter.Label(my_w,  text='Item : ',
width=10,anchor="c" )
```

```
    l3.grid(row=5,column=1)
    tItemPay = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tItemPay.grid(row=5,column=2)

    b1 = tkinter.Button(my_w,  text='Pay', width=10,
command=lambda: pay())
    b1.grid(row=7,column=2)
    print('Make payment')

def pay() :
    my_name = tNamePay.get("1.0",END)
    my_phone = tPhonePay.get("1.0",END)

    query="DELETE FROM `Payment` WHERE Name = %s AND Phone = %s"
    myCursor.execute(query,(my_name, my_phone,))
    db_connection.commit()
    tNamePay.delete('1.0',END)
    tPhonePay.delete('1.0',END)
    tAddressPay.delete('1.0',END)
    tItemPay.delete('1.0',END)
    print("Query executed")
```

2. Takes

```
def TakePayments() :
    global tNameReq, tPhoneReq, tAddressReq, tItemReq

    my_w = tkinter.Tk()
    my_w.geometry("300x300")
    my_w.title("Request Payment")
    l0 = tkinter.Label(my_w,  text='Request
Payments',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Customer Name : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tNameReq = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tNameReq.grid(row=3,column=2)
    l2 = tkinter.Label(my_w,  text='Customer Phone: ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tPhoneReq = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tPhoneReq.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Customer Address : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    tAddressReq = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tAddressReq.grid(row=5,column=2)

    l4 = tkinter.Label(my_w,  text='Ordered Item : ',
width=10,anchor="c" )
    l4.grid(row=6,column=1)
    tItemReq = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tItemReq.grid(row=6,column=2)

    b1 = tkinter.Button(my_w,  text='Put the item on sale',
width=10, command=lambda: requestPayment())
    b1.grid(row=7,column=2)

def requestPayment() :
    my_name = tNameReq.get("1.0",END)
    my_class = tPhoneReq.get("1.0",END)
    my_mark = tAddressReq.get("1.0",END)
    my_gender = tItemReq.get("1.0",END)
    query="INSERT INTO  `Payment`
(`Name` ,`Phone` ,`Address` ,`Item`) VALUES (%s,%s,%s,%s)"
    my_data=(my_name,my_class,my_mark,my_gender)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tNameReq.delete('1.0',END)
    tPhoneReq.delete('1.0',END)
    tAddressReq.delete('1.0',END)
    tItemReq.delete('1.0',END)
    print("Query executed")
```

Orders

1. Places

```
def PlaceOrder() :
    global OrderCname, OrderPhone, OrderAddress, OrderAmount,
OrderProject, OrderItem

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Place Order")
    l0 = tkinter.Label(my_w,  text='Place
Order',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Customer Name : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    OrderCname = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderCname.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Phone : ', width=10,anchor="c"
)
    l2.grid(row=4,column=1)
    OrderPhone = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderPhone.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Address : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    OrderAddress = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderAddress.grid(row=5,column=2)

    l4 = tkinter.Label(my_w,  text='Amount : ',
width=10,anchor="c" )
    l4.grid(row=6,column=1)
    OrderAmount = tkinter.Text(my_w,  height=1,
width=10,bg='white')
```

```
    OrderAmount.grid(row=6,column=2)

    l5 = tkinter.Label(my_w,  text='Project : ',
width=10,anchor="c" )
    l5.grid(row=7,column=1)
    OrderProject = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderProject.grid(row=7,column=2)

    l6 = tkinter.Label(my_w,  text='Item : ',
width=10,anchor="c" )
    l6.grid(row=8,column=1)
    OrderItem = tkinter.Text(my_w,  height=1, width=10,bg='white')
    OrderItem.grid(row=8,column=2)

    b1 = tkinter.Button(my_w,  text='Place Order', width=10,
command=lambda: give_order())
    b1.grid(row=10,column=2)
    print('Place order')

def give_order() :
    Cname = OrderCname.get('1.0',END)
    Phone = OrderPhone.get('1.0',END)
    Address = OrderAddress.get('1.0',END)
    Amount = OrderAmount.get('1.0',END)
    Project = OrderProject.get('1.0',END)
    Item = OrderItem.get('1.0',END)

    query = """INSERT INTO OrderTable VALUES (%s,%s,%s,%s,%s,
%s)"""
    myData = (Cname, Phone, Address, Amount, Project, Item)
    myCursor.execute(query, myData)

    db_connection.commit()

    OrderCname.delete('1.0', END)
    OrderPhone.delete('1.0', END)
    OrderAddress.delete('1.0', END)
    OrderAmount.delete('1.0', END)
    OrderProject.delete('1.0', END)
    OrderItem.delete('1.0', END)

    print('Give order')
```

2. Completes

```
def CompleteOrder() :

    global CompCustName, CompCustPhone

    myCursor.execute("SELECT * FROM OrderTable")
    my_wo = tkinter.Tk()
    my_wo.title("Requested orders ")
    my_wo.geometry("250x250")
    i=0
    for Project in myCursor:
        for j in range(len(Project)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Project[j])
        i=i+1
    print('Complete order')

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Complete Order")
    l0 = tkinter.Label(my_w,  text='Complete 
Order',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Customer Name : ', 
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    CompCustName = tkinter.Text(my_w,  height=1, 
width=10,bg='white')
    CompCustName.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Phone : ', width=10,anchor="c" 
)
    l2.grid(row=4,column=1)
    CompCustPhone = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    CompCustPhone.grid(row=4,column=2)

    b1 = tkinter.Button(my_w,  text='Complete Order', width=40,
command=lambda: comp_the_order())
    b1.grid(row=7,column=2)

def comp_the_order() :
    CustName = CompCustName.get('1.0', END)
    CustPhone = CompCustPhone.get('1.0', END)

    query = """DELETE FROM OrderTable WHERE Phone = %s"""
    myData = (CustPhone,)

    myCursor.execute(query, myData)
    db_connection.commit()

    CompCustName.delete('1.0', END)
    CompCustPhone.delete('1.0', END)
```

Project

1. **Gives**

```
def GiveProject() :
    global tGiveProj, tGiveName, tGiveNum, tGiveVend, tGiveItem,
tGiveDetails

    my_w = tkinter.Tk()
    my_w.geometry("300x300")
    my_w.title("Give Project")
    l0 = tkinter.Label(my_w,  text='Give
Project',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Project Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tGiveProj = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveProj.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Customer Name : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tGiveName = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveName.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Customer Number : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    tGiveNum = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveNum.grid(row=5,column=2)

    l4 = tkinter.Label(my_w,  text='Vendor Number : ',
width=10,anchor="c" )
    l4.grid(row=6,column=1)
    tGiveVend = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveVend.grid(row=6,column=2)

    l5 = tkinter.Label(my_w,  text='Item : ',
width=10,anchor="c" )
    l5.grid(row=7,column=1)
    tGiveItem = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveItem.grid(row=7,column=2)

    l6 = tkinter.Label(my_w,  text='Details : ',
width=10,anchor="c" )
    l6.grid(row=8,column=1)
    tGiveDetails = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tGiveDetails.grid(row=8,column=2)

    b1 = tkinter.Button(my_w,  text='Give Project', width=10,
command=lambda: giveTheProject())
    b1.grid(row=10,column=2)

    print('Give project')

def giveTheProject() :
    my_name = tGiveProj.get("1.0",END)
    my_class = tGiveName.get("1.0",END)
    my_mark = tGiveNum.get("1.0",END)
    my_gender = tGiveVend.get("1.0",END)
    my_item = tGiveItem.get("1.0",END)
    my_details = tGiveDetails.get("1.0",END)
    query="INSERT INTO  `Project` (`ProjectNumber` ,`CustomerName`
,`CustomerNumber` ,`VendorNumber`, `Item`, `Details`) VALUES (%s,
%s,%s,%s,%s,%s)"
    my_data=(my_name,my_class,my_mark,my_gender, my_item,
my_details)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tGiveProj.delete('1.0',END)
    tGiveName.delete('1.0',END)
    tGiveNum.delete('1.0',END)
    tGiveVend.delete('1.0',END)
    tGiveItem.delete('1.0',END)
    tGiveDetails.delete('1.0',END)
    print("Query executed")
```

2. Takes

```
def TakesProjects() :
    global tTakeProj, tTakeName, tTakeDetails
    myCursor.execute("SELECT * FROM Project")
    my_wo = tkinter.Tk()
    my_wo.title("Requested projects ")
    my_wo.geometry("250x250")
    i=0
    for Project in myCursor:
        for j in range(len(Project)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Project[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Take Project")
    l0 = tkinter.Label(my_w,  text='Take 
Project',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Project Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tTakeProj = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tTakeProj.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Vendor Name : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tTakeName = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tTakeName.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Give details : ',
width=10,anchor="c" )
    l3.grid(row=4,column=1)
    tTakeDetails = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tTakeDetails.grid(row=4,column=2)

    b1 = tkinter.Button(my_w,  text='Take the project', width=10,
command=lambda: taketheProj())
    b1.grid(row=6,column=2)

def taketheProj() :
    my_class = tTakeProj.get("1.0",END)
    my_mark = tTakeDetails.get("1.0",END)
    query = """UPDATE Project SET Details = %s WHERE ProjectNumber
= %s"""
    my_data = (my_mark,my_class,)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tTakeName.delete('1.0',END)
    tTakeDetails.delete('1.0',END)
    tTakeProj.delete('1.0',END)
    print("Query executed")
```

3. Completes

```
def CompletesProjects() :
    global tCompProj, tCompName, tCompDetails
    myCursor.execute("SELECT * FROM Project")
    my_wo = tkinter.Tk()
    my_wo.title("Requested projects ")
    my_wo.geometry("250x250")
    i=0
    for Project in myCursor:
        for j in range(len(Project)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Project[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Take Project")
    l0 = tkinter.Label(my_w,  text='Project
Completed!',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Project Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tCompProj = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tCompProj.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Vendor Name : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tCompName = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tCompName.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Give details : ',
width=10,anchor="c" )
    l3.grid(row=4,column=1)
    tCompDetails = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tCompDetails.grid(row=4,column=2)
    b1 = tkinter.Button(my_w,  text='Completed', width=10,
command=lambda: compTheProj())
    b1.grid(row=6,column=2)

def compTheProj() :
    my_class = tCompProj.get("1.0",END)
    query = """DELETE FROM Project WHERE ProjectNumber = %s"""
    my_data = (my_class,)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tCompName.delete('1.0',END)
    tCompDetails.delete('1.0',END)
    tCompProj.delete('1.0',END)
    print("Query executed")
```

Complete Code

```
import tkinter
from tkinter import *
import mysql.connector

db_connection = mysql.connector.connect(host = '127.0.0.1', user =
'root', password = 'Chiku$!
d9',auth_plugin='mysql_native_password', database = 'Goconstruct')
myCursor = db_connection.cursor()

tItem = None
tQuantity = None
tOT = None
tNamePay = None
tPhonePay = None
tAddressPay = None
tItemPay = None
OrderCname = None
OrderPhone = None
OrderAddress = None
OrderAmount = None
OrderProject = None
```

```
OrderItem = None
tGiveProj = None
tGiveName = None
tGiveNum = None
tGiveVend = None
tGiveItem = None
tGiveDetails = None
tNameReq = None
tPhoneReq = None
tAddressReq = None
tItemReq = None
t1 = None
t2 = None
t3 = None
t4 = None
tCompProj = None
tCompName = None
tCompDetails = None
tTakeProj = None
tTakeName = None
tTakeDetails = None
CompCustName = None
CompCustPhone = None

main = tkinter.Tk()
main.title('Entry Window')
main.configure(bg='wheat')
main.geometry("200x200")

def CustomerDial() :
    Customer = tkinter.Tk()
    Customer.configure(bg='wheat')
    Customer.title('Customer Functions')
    NeedsItems = tkinter.Button(Customer, text = "Needs Items",
command = NeedItem)
    MakesPayments = tkinter.Button(Customer, text = "Makes
Payments", command = MakePayment)
    PlacesOrders = tkinter.Button(Customer, text = "Place Orders",
command = PlaceOrder)
```

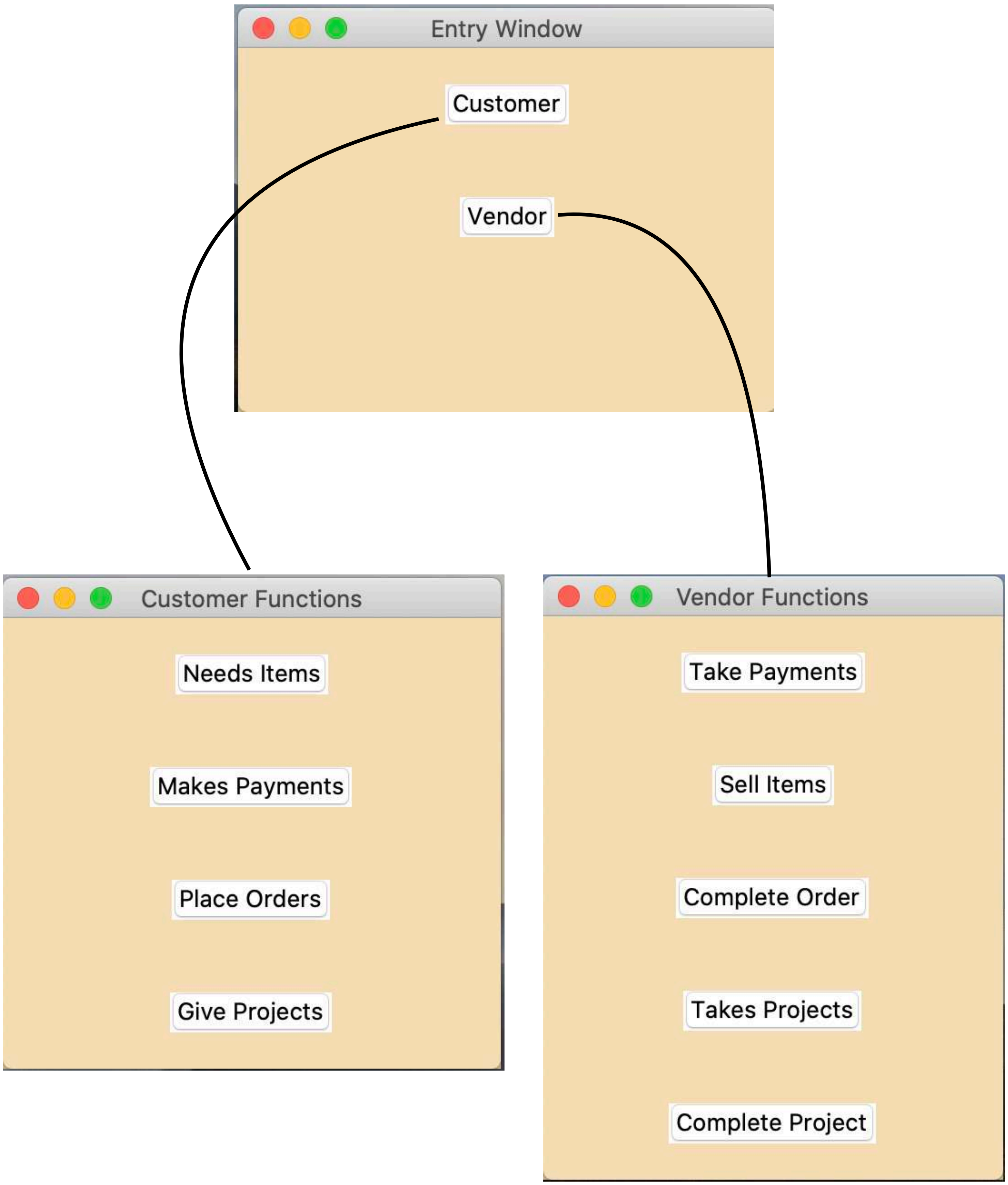
```
    GivesProjects = tkinter.Button(Customer, text = "Give
Projects", command = GiveProject)
    NeedsItems.pack(padx=20, pady=20)
    MakesPayments.pack(padx=20, pady=20)
    PlacesOrders.pack(padx=20, pady=20)
    GivesProjects.pack(padx=20, pady=20)

def NeedItem() :
    global tItem, tQuantity, tOT

    myCursor.execute("SELECT * FROM Item")
    my_wo = tkinter.Tk()
    my_wo.title("Available Items")
    my_wo.geometry("250x250")
    i=0
    for Item in myCursor:
        for j in range(len(Item)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Item[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Buy Items")
    l0 = tkinter.Label(my_w,  text='Needs
Items',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Item Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tItem = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tItem.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Quantity : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tQuantity = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tQuantity.grid(row=4,column=2)
    l3 = tkinter.Label(my_w,  text='Order : ', width=10,anchor="c"
)
    l3.grid(row=5,column=1)
    tOT = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tOT.grid(row=5,column=2)

    b1 = tkinter.Button(my_w,  text='Buy', width=10,
command=lambda: delete_data_item())
    b1.grid(row=7,column=2)

def delete_data_item() :
    my_name = tItem.get("1.0",END)
    query="DELETE FROM `Item` WHERE ItemNumber = %s"
    myCursor.execute(query,(my_name,))
    db_connection.commit()
    tItem.delete('1.0',END)
    tQuantity.delete('1.0',END)
    tOT.delete('1.0',END)
    print("Query executed")

def MakePayment() :
    global tNamePay, tPhonePay, tAddressPay, tItemPay

    myCursor.execute("SELECT * FROM Payment")
    my_wo = tkinter.Tk()
    my_wo.title("Requested payments ")
    my_wo.geometry("250x250")
    i=0
    for Item in myCursor:
        for j in range(len(Item)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Item[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Pay")
    l0 = tkinter.Label(my_w,  text='Make
Payment',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Name : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tNamePay = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tNamePay.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Phone : ', width=10,anchor="c"
)
    l2.grid(row=4,column=1)
    tPhonePay = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tPhonePay.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Address : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    tAddressPay = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tAddressPay.grid(row=5,column=2)

    l3 = tkinter.Label(my_w,  text='Item : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    tItemPay = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tItemPay.grid(row=5,column=2)

    b1 = tkinter.Button(my_w,  text='Pay', width=10,
command=lambda: pay())
    b1.grid(row=7,column=2)
    print('Make payment')

def pay() :
    my_name = tNamePay.get("1.0",END)
    my_phone = tPhonePay.get("1.0",END)

    query="DELETE FROM `Payment` WHERE Name = %s AND Phone = %s"
    myCursor.execute(query,(my_name, my_phone,))
    db_connection.commit()
    tNamePay.delete('1.0',END)
    tPhonePay.delete('1.0',END)
    tAddressPay.delete('1.0',END)
    tItemPay.delete('1.0',END)
    print("Query executed")

def PlaceOrder() :
    global OrderCname, OrderPhone, OrderAddress, OrderAmount,
OrderProject, OrderItem

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Place Order")
    l0 = tkinter.Label(my_w,  text='Place
Order',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Customer Name : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    OrderCname = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderCname.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Phone : ', width=10,anchor="c"
)
    l2.grid(row=4,column=1)
    OrderPhone = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderPhone.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Address : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    OrderAddress = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderAddress.grid(row=5,column=2)
    l4 = tkinter.Label(my_w,  text='Amount : ',
width=10,anchor="c" )
    l4.grid(row=6,column=1)
    OrderAmount = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderAmount.grid(row=6,column=2)

    l5 = tkinter.Label(my_w,  text='Project : ',
width=10,anchor="c" )
    l5.grid(row=7,column=1)
    OrderProject = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    OrderProject.grid(row=7,column=2)

    l6 = tkinter.Label(my_w,  text='Item : ',
width=10,anchor="c" )
    l6.grid(row=8,column=1)
    OrderItem = tkinter.Text(my_w,  height=1, width=10,bg='white')
    OrderItem.grid(row=8,column=2)

    b1 = tkinter.Button(my_w,  text='Place Order', width=10,
command=lambda: give_order())
    b1.grid(row=10,column=2)
    print('Place order')

def give_order() :
    Cname = OrderCname.get('1.0',END)
    Phone = OrderPhone.get('1.0',END)
    Address = OrderAddress.get('1.0',END)
    Amount = OrderAmount.get('1.0',END)
    Project = OrderProject.get('1.0',END)
    Item = OrderItem.get('1.0',END)

    query = """INSERT INTO OrderTable VALUES (%s,%s,%s,%s,%s,
%s)"""
    myData = (Cname, Phone, Address, Amount, Project, Item)
    myCursor.execute(query, myData)

    db_connection.commit()
    OrderCname.delete('1.0', END)
    OrderPhone.delete('1.0', END)
    OrderAddress.delete('1.0', END)
    OrderAmount.delete('1.0', END)
    OrderProject.delete('1.0', END)
    OrderItem.delete('1.0', END)

    print('Give order')

def GiveProject() :
    global tGiveProj, tGiveName, tGiveNum, tGiveVend, tGiveItem,
tGiveDetails

    my_w = tkinter.Tk()
    my_w.geometry("300x300")
    my_w.title("Give Project")
    l0 = tkinter.Label(my_w,  text='Give
Project',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Project Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tGiveProj = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveProj.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Customer Name : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tGiveName = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveName.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Customer Number : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    tGiveNum = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveNum.grid(row=5,column=2)

    l4 = tkinter.Label(my_w,  text='Vendor Number : ',
width=10,anchor="c" )
```

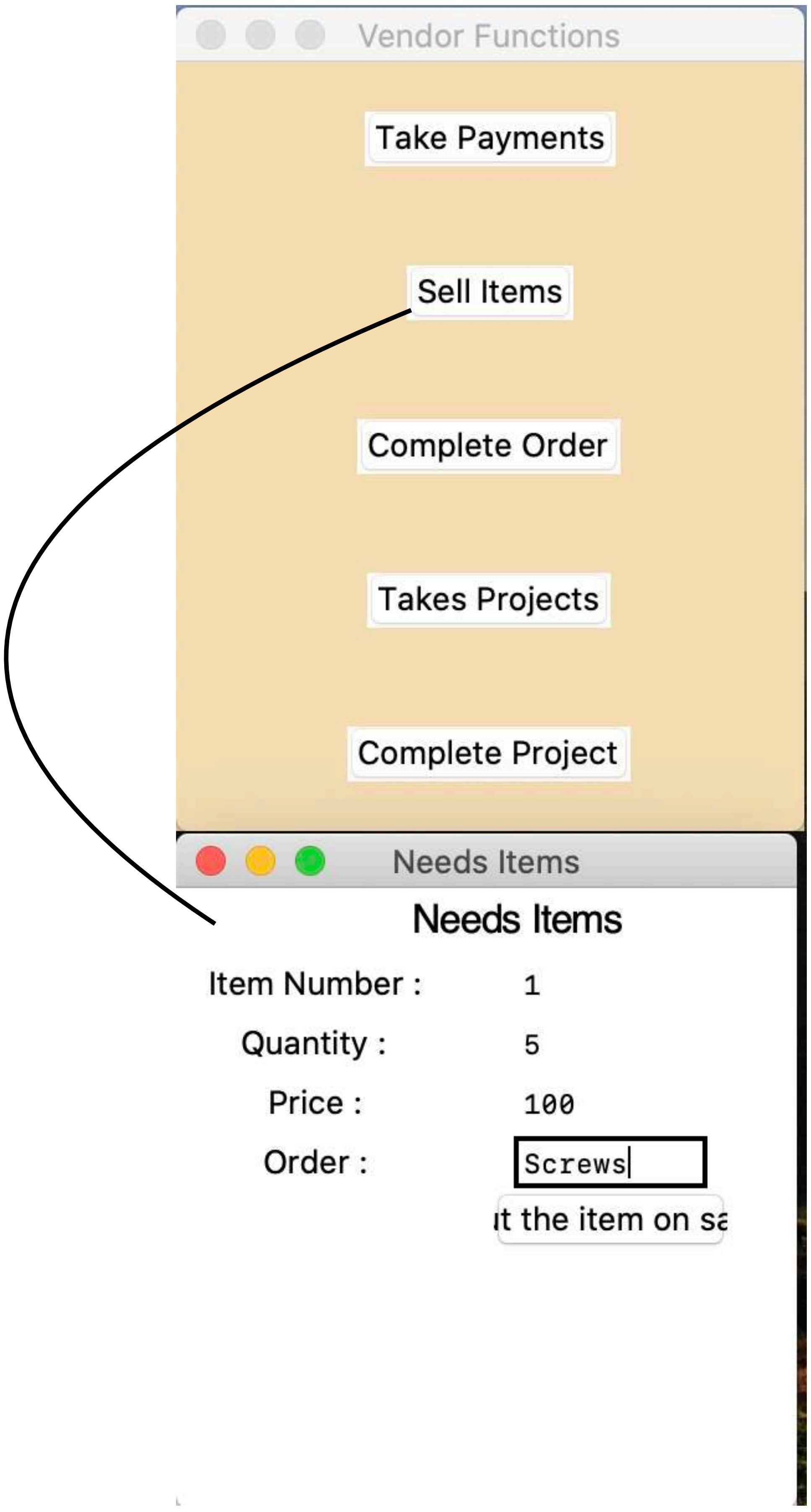
```
    l4.grid(row=6,column=1)
    tGiveVend = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveVend.grid(row=6,column=2)

    l5 = tkinter.Label(my_w,  text='Item : ',
width=10,anchor="c" )
    l5.grid(row=7,column=1)
    tGiveItem = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tGiveItem.grid(row=7,column=2)

    l6 = tkinter.Label(my_w,  text='Details : ',
width=10,anchor="c" )
    l6.grid(row=8,column=1)
    tGiveDetails = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tGiveDetails.grid(row=8,column=2)

    b1 = tkinter.Button(my_w,  text='Give Project', width=10,
command=lambda: giveTheProject())
    b1.grid(row=10,column=2)

    print('Give project')

def giveTheProject() :
    my_name = tGiveProj.get("1.0",END)
    my_class = tGiveName.get("1.0",END)
    my_mark = tGiveNum.get("1.0",END)
    my_gender = tGiveVend.get("1.0",END)
    my_item = tGiveItem.get("1.0",END)
    my_details = tGiveDetails.get("1.0",END)
    query="INSERT INTO  `Project` (`ProjectNumber` ,`CustomerName`
,`CustomerNumber` ,`VendorNumber`, `Item`, `Details`) VALUES (%s,
%s,%s,%s,%s,%s)"
    my_data=(my_name,my_class,my_mark,my_gender, my_item,
my_details)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tGiveProj.delete('1.0',END)
    tGiveName.delete('1.0',END)
    tGiveNum.delete('1.0',END)
    tGiveVend.delete('1.0',END)
    tGiveItem.delete('1.0',END)
    tGiveDetails.delete('1.0',END)
    print("Query executed")

def VendorDial() :
    Vendor = tkinter.Tk()
    Vendor.configure(bg='wheat')
    Vendor.title('Vendor Functions')
    TakesPayment = tkinter.Button(Vendor, text = "Take Payments",
command = TakePayments)
    SellsItem = tkinter.Button(Vendor, text = "Sell Items",
command = SellsItems)
    CompletesOrder = tkinter.Button(Vendor, text = "Complete
Order", command = CompleteOrder)
    TakesProject = tkinter.Button(Vendor, text = "Takes Projects",
command = TakesProjects)
    CompletesProject = tkinter.Button(Vendor, text = "Complete
Project", command = CompletesProjects)
    TakesPayment.pack(padx=20, pady=20)
    SellsItem.pack(padx=20, pady=20)
    CompletesOrder.pack(padx=20, pady=20)
    TakesProject.pack(padx=20, pady=20)
    CompletesProject.pack(padx=20, pady=20)

def TakePayments() :
    global tNameReq, tPhoneReq, tAddressReq, tItemReq

    my_w = tkinter.Tk()
    my_w.geometry("300x300")
    my_w.title("Request Payment")
    l0 = tkinter.Label(my_w,  text='Request
Payments',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Customer Name : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tNameReq = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tNameReq.grid(row=3,column=2)
    l2 = tkinter.Label(my_w,  text='Customer Phone: ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tPhoneReq = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tPhoneReq.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Customer Address : ',
width=10,anchor="c" )
    l3.grid(row=5,column=1)
    tAddressReq = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tAddressReq.grid(row=5,column=2)

    l4 = tkinter.Label(my_w,  text='Ordered Item : ',
width=10,anchor="c" )
    l4.grid(row=6,column=1)
    tItemReq = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tItemReq.grid(row=6,column=2)

    b1 = tkinter.Button(my_w,  text='Put the item on sale',
width=10, command=lambda: requestPayment())
    b1.grid(row=7,column=2)

def requestPayment() :
    my_name = tNameReq.get("1.0",END)
    my_class = tPhoneReq.get("1.0",END)
    my_mark = tAddressReq.get("1.0",END)
    my_gender = tItemReq.get("1.0",END)
    query="INSERT INTO  `Payment`
(`Name` ,`Phone` ,`Address` ,`Item`) VALUES (%s,%s,%s,%s)"
    my_data=(my_name,my_class,my_mark,my_gender)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tNameReq.delete('1.0',END)
    tPhoneReq.delete('1.0',END)
    tAddressReq.delete('1.0',END)
    tItemReq.delete('1.0',END)
    print("Query executed")
```

```
def SellsItems() :
    global t1, t2, t3, t4

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Needs Items")
    l0 = tkinter.Label(my_w,  text='Needs
Items',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Item Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    t1 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t1.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Quantity : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    t2 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t2.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Price : ', width=10,anchor="c"
)
    l3.grid(row=5,column=1)
    t3 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t3.grid(row=5,column=2)

    l4 = tkinter.Label(my_w,  text='Order : ', width=10,anchor="c"
)
    l4.grid(row=6,column=1)
    t4 = tkinter.Text(my_w,  height=1, width=10,bg='white')
    t4.grid(row=6,column=2)

    b1 = tkinter.Button(my_w,  text='Put the item on sale',
width=10, command=lambda: add_data_item())
    b1.grid(row=7,column=2)

def add_data_item() :
    my_name = t1.get("1.0",END)
    my_class = t2.get("1.0",END)
    my_mark = t3.get("1.0",END)
    my_gender = t4.get("1.0",END)
    query="INSERT INTO  `Item`
(`ItemNumber` ,`Quantity` ,`Price` ,`OrderItem`) VALUES(%s,%s,%s,
%s)"
    my_data=(my_name,my_class,my_mark,my_gender)
    myCursor.execute(query,my_data)
    db_connection.commit()
    t1.delete('1.0',END)
    t2.delete('1.0',END)
    t3.delete('1.0',END)
    t4.delete('1.0',END)
    print("Query executed")

def CompleteOrder() :

    global CompCustName, CompCustPhone

    myCursor.execute("SELECT * FROM OrderTable")
    my_wo = tkinter.Tk()
    my_wo.title("Requested orders ")
    my_wo.geometry("250x250")
    i=0
    for Project in myCursor:
        for j in range(len(Project)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Project[j])
        i=i+1
    print('Complete order')

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Complete Order")
    l0 = tkinter.Label(my_w,  text='Complete
Order',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)
    l1 = tkinter.Label(my_w,  text='Customer Name : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    CompCustName = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    CompCustName.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Phone : ', width=10,anchor="c"
)
    l2.grid(row=4,column=1)
    CompCustPhone = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    CompCustPhone.grid(row=4,column=2)

    b1 = tkinter.Button(my_w,  text='Complete Order', width=40,
command=lambda: comp_the_order())
    b1.grid(row=7,column=2)

def comp_the_order() :
    CustName = CompCustName.get('1.0', END)
    CustPhone = CompCustPhone.get('1.0', END)

    query = """DELETE FROM OrderTable WHERE Phone = %s"""
    myData = (CustPhone,)

    myCursor.execute(query, myData)
    db_connection.commit()

    CompCustName.delete('1.0', END)
    CompCustPhone.delete('1.0', END)

def TakesProjects() :
    global tTakeProj, tTakeName, tTakeDetails
    myCursor.execute("SELECT * FROM Project")
    my_wo = tkinter.Tk()
    my_wo.title("Requested projects ")
    my_wo.geometry("250x250")
    i=0
    for Project in myCursor:
        for j in range(len(Project)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Project[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Take Project")
    l0 = tkinter.Label(my_w,  text='Take
Project',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Project Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tTakeProj = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tTakeProj.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Vendor Name : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tTakeName = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tTakeName.grid(row=4,column=2)

    l3 = tkinter.Label(my_w,  text='Give details : ',
width=10,anchor="c" )
    l3.grid(row=4,column=1)
    tTakeDetails = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tTakeDetails.grid(row=4,column=2)

    b1 = tkinter.Button(my_w,  text='Take the project', width=10,
command=lambda: taketheProj())
    b1.grid(row=6,column=2)

def taketheProj() :
    my_class = tTakeProj.get("1.0",END)
    my_mark = tTakeDetails.get("1.0",END)
    query = """UPDATE Project SET Details = %s WHERE ProjectNumber
= %s"""
```

```
    my_data = (my_mark,my_class,)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tTakeName.delete('1.0',END)
    tTakeDetails.delete('1.0',END)
    tTakeProj.delete('1.0',END)
    print("Query executed")

def CompletesProjects() :
    global tCompProj, tCompName, tCompDetails
    myCursor.execute("SELECT * FROM Project")
    my_wo = tkinter.Tk()
    my_wo.title("Requested projects ")
    my_wo.geometry("250x250")
    i=0
    for Project in myCursor:
        for j in range(len(Project)):
            e = Entry(my_wo, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, Project[j])
        i=i+1

    my_w = tkinter.Tk()
    my_w.geometry("250x250")
    my_w.title("Take Project")
    l0 = tkinter.Label(my_w,  text='Project
Completed!',font=('Helvetica', 16), width=30,anchor="c" )
    l0.grid(row=1,column=1,columnspan=4)

    l1 = tkinter.Label(my_w,  text='Project Number : ',
width=10,anchor="c" )
    l1.grid(row=3,column=1)
    tCompProj = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tCompProj.grid(row=3,column=2)

    l2 = tkinter.Label(my_w,  text='Vendor Name : ',
width=10,anchor="c" )
    l2.grid(row=4,column=1)
    tCompName = tkinter.Text(my_w,  height=1, width=10,bg='white')
    tCompName.grid(row=4,column=2)
    l3 = tkinter.Label(my_w,  text='Give details : ',
width=10,anchor="c" )
    l3.grid(row=4,column=1)
    tCompDetails = tkinter.Text(my_w,  height=1,
width=10,bg='white')
    tCompDetails.grid(row=4,column=2)

    b1 = tkinter.Button(my_w,  text='Completed', width=10,
command=lambda: compTheProj())
    b1.grid(row=6,column=2)

def compTheProj() :
    my_class = tCompProj.get("1.0",END)
    query = """DELETE FROM Project WHERE ProjectNumber = %s"""
    my_data = (my_class,)
    myCursor.execute(query,my_data)
    db_connection.commit()
    tCompName.delete('1.0',END)
    tCompDetails.delete('1.0',END)
    tCompProj.delete('1.0',END)
    print("Query executed")

def HelloCallBack() :
    C = tkinter.Button(main, text = "Customer", command =
CustomerDial, compound = "c")
    V = tkinter.Button(main, text = "Vendor", command =
VendorDial, compound = "c")
    C.pack(padx=20, pady=20)
    V.pack(padx=20, pady=20)

HelloCallBack()

main.mainloop()
```

Simulated screenshots





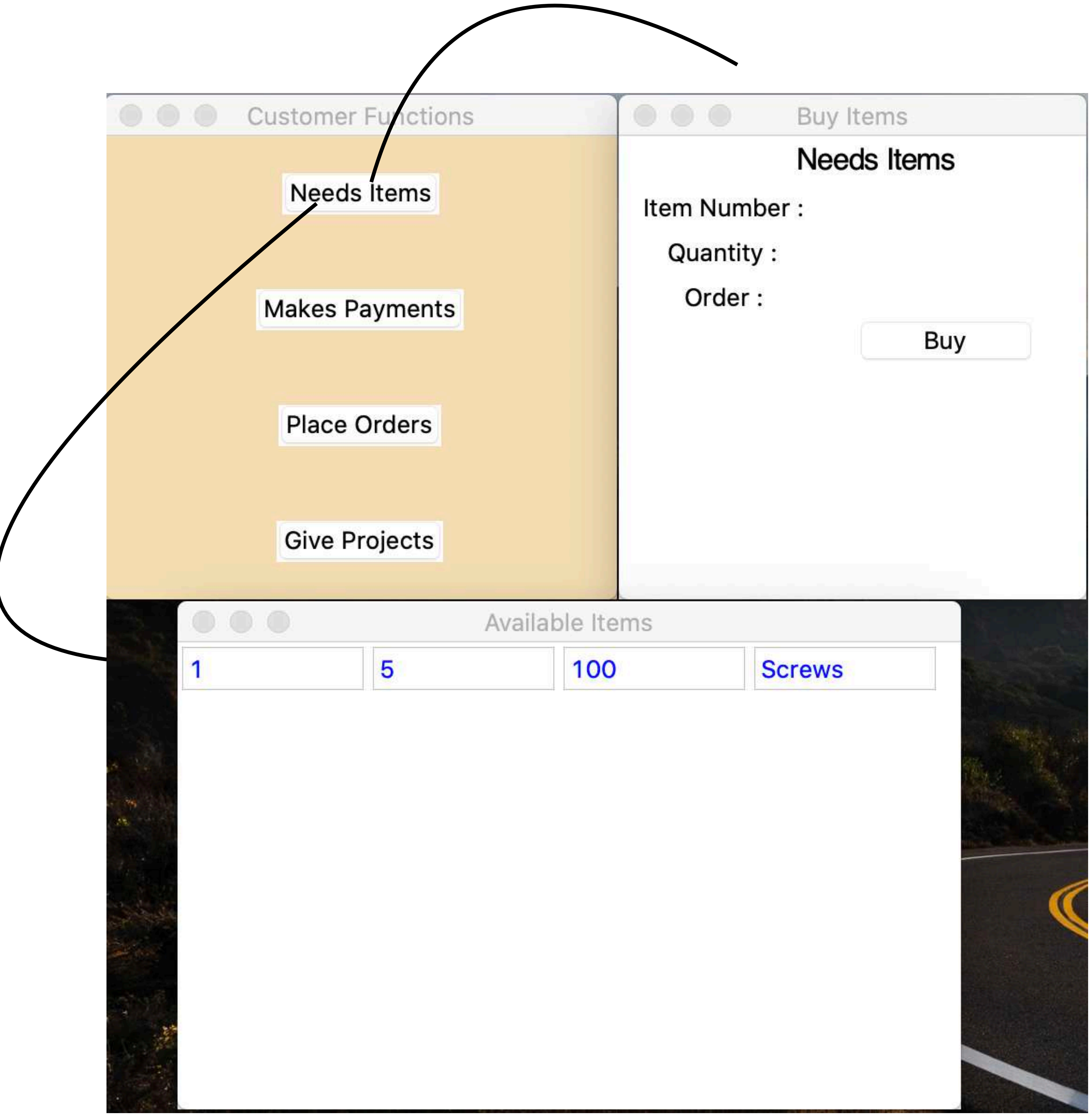
Needs Items

Item Number : 1

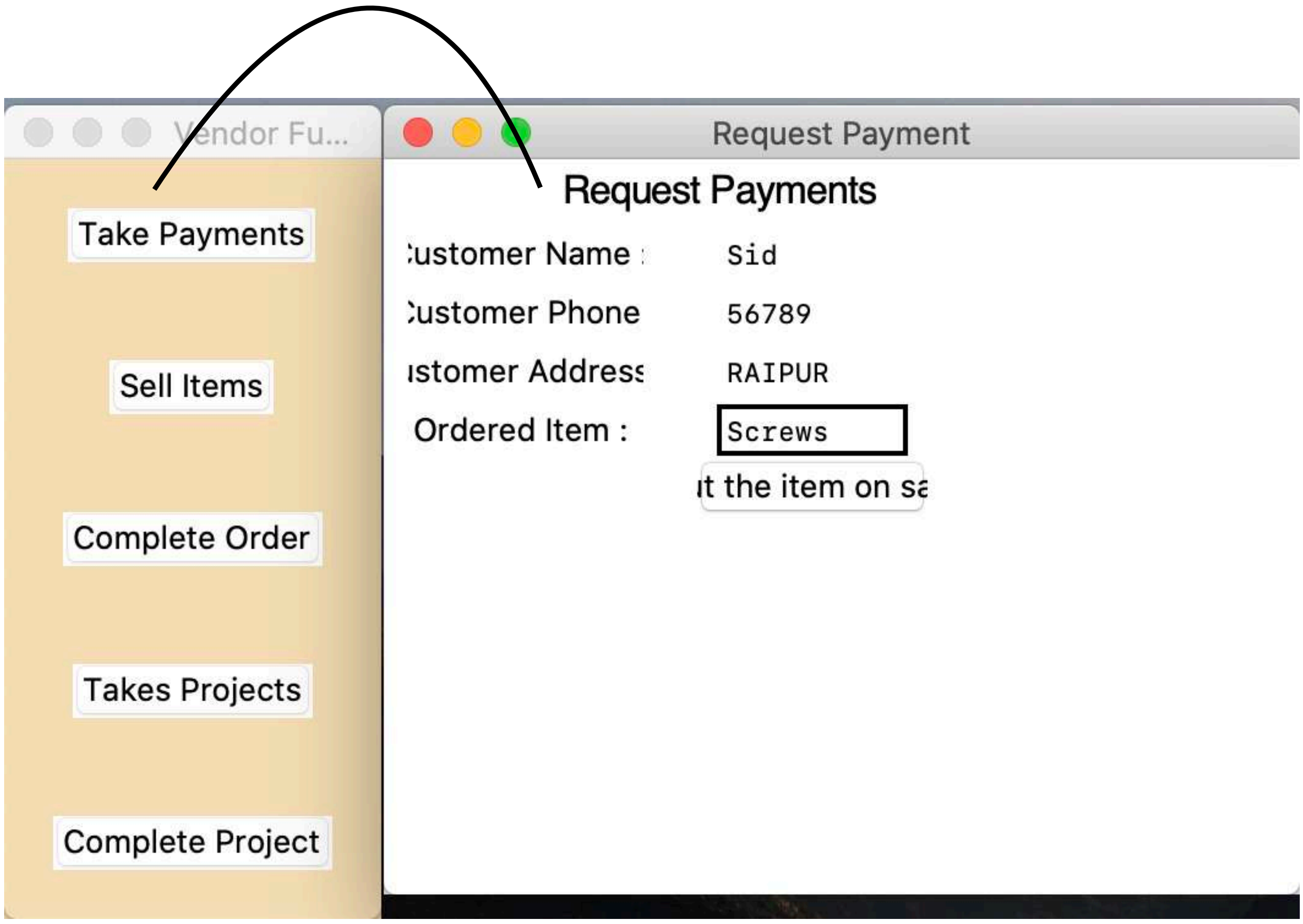
Quantity : 5

Price : 100

Order : Screws



1	5	100	Screws



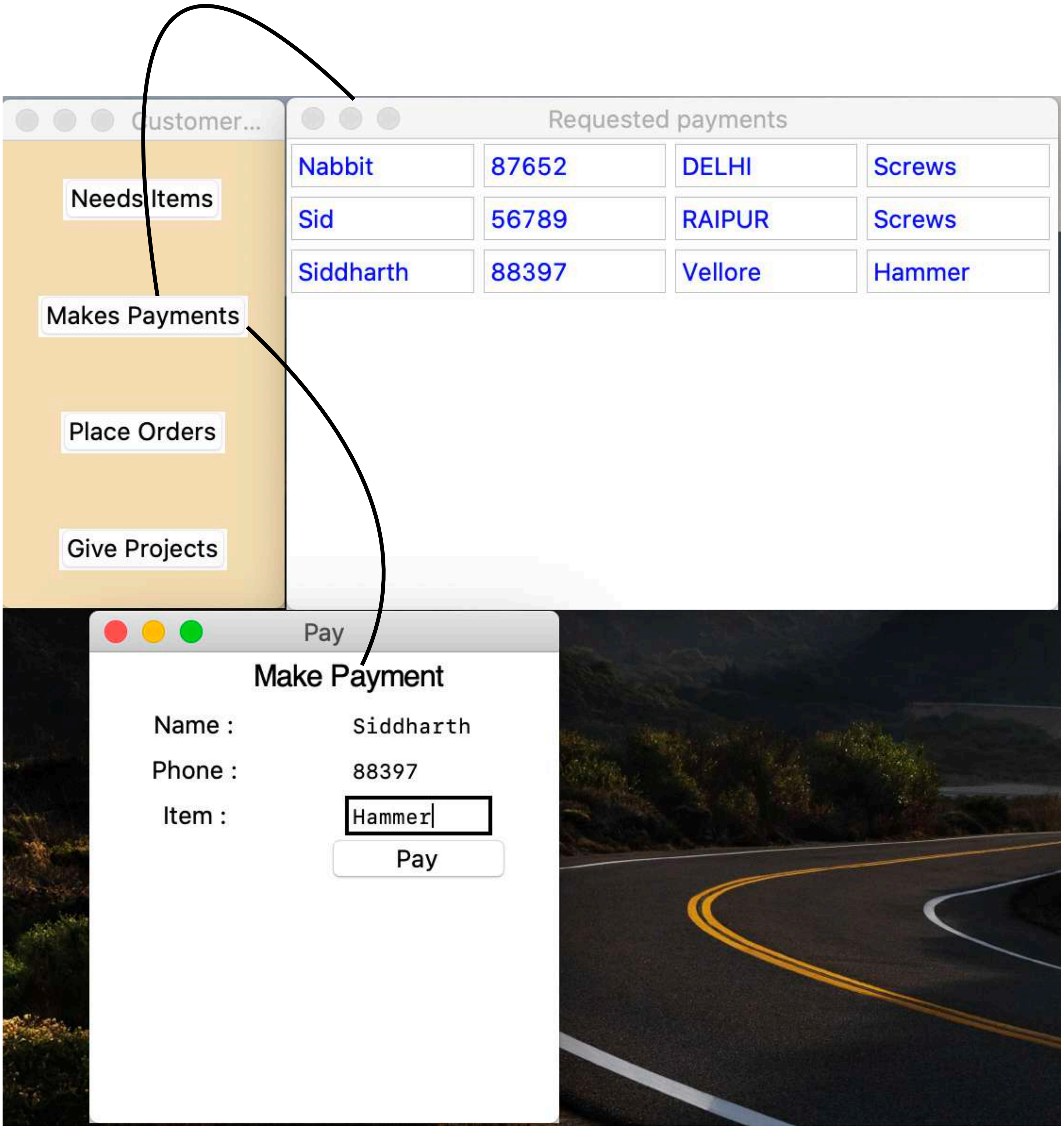
Request Payments

:ustomer Name : Sid

:ustomer Phone 56789

ıstomer Address RAIPUR

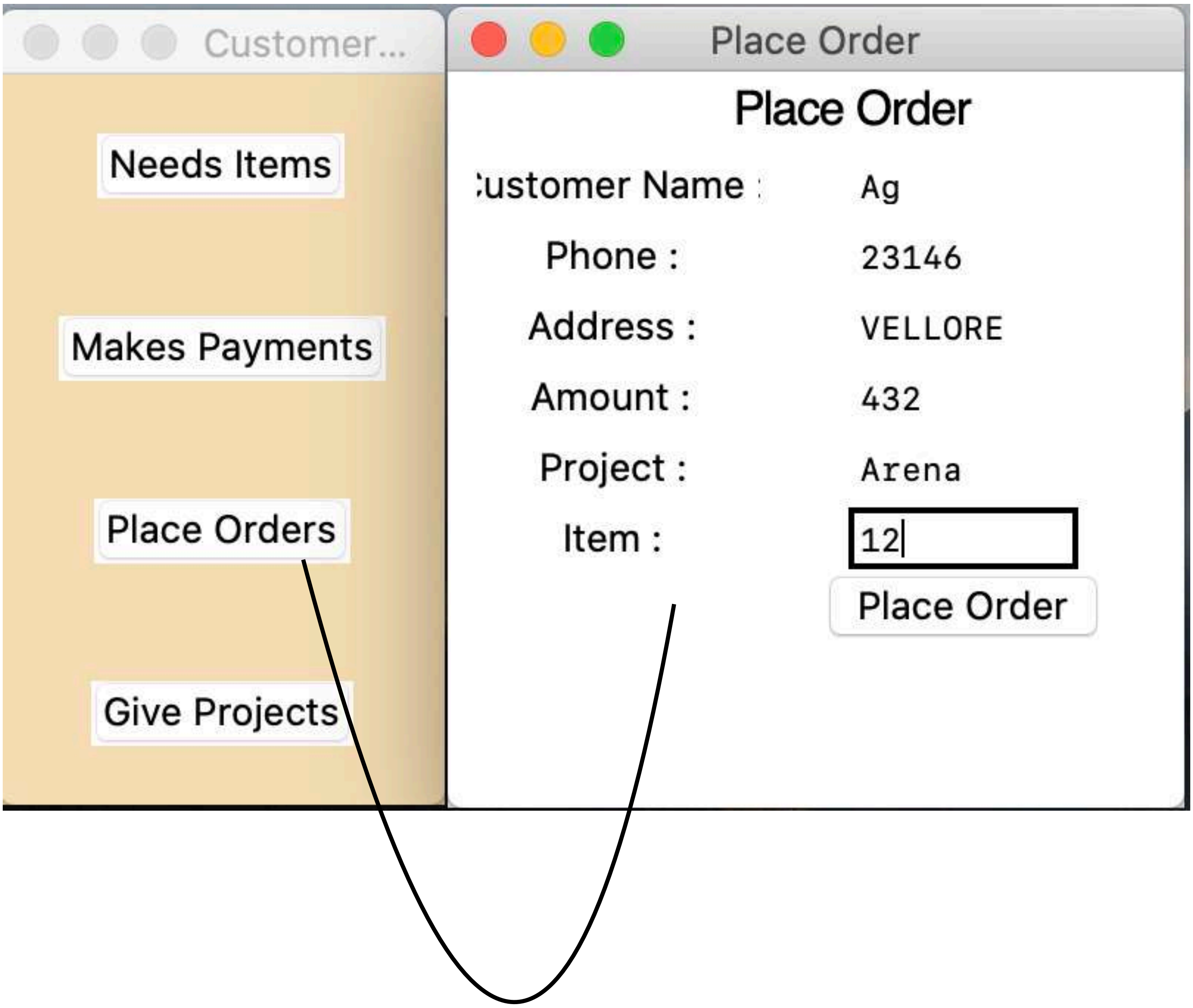
Ordered Item : Screws



Nabbit	87652	DELHI	Screws
Sid	56789	RAIPUR	Screws
Siddharth	88397	Vellore	Hammer

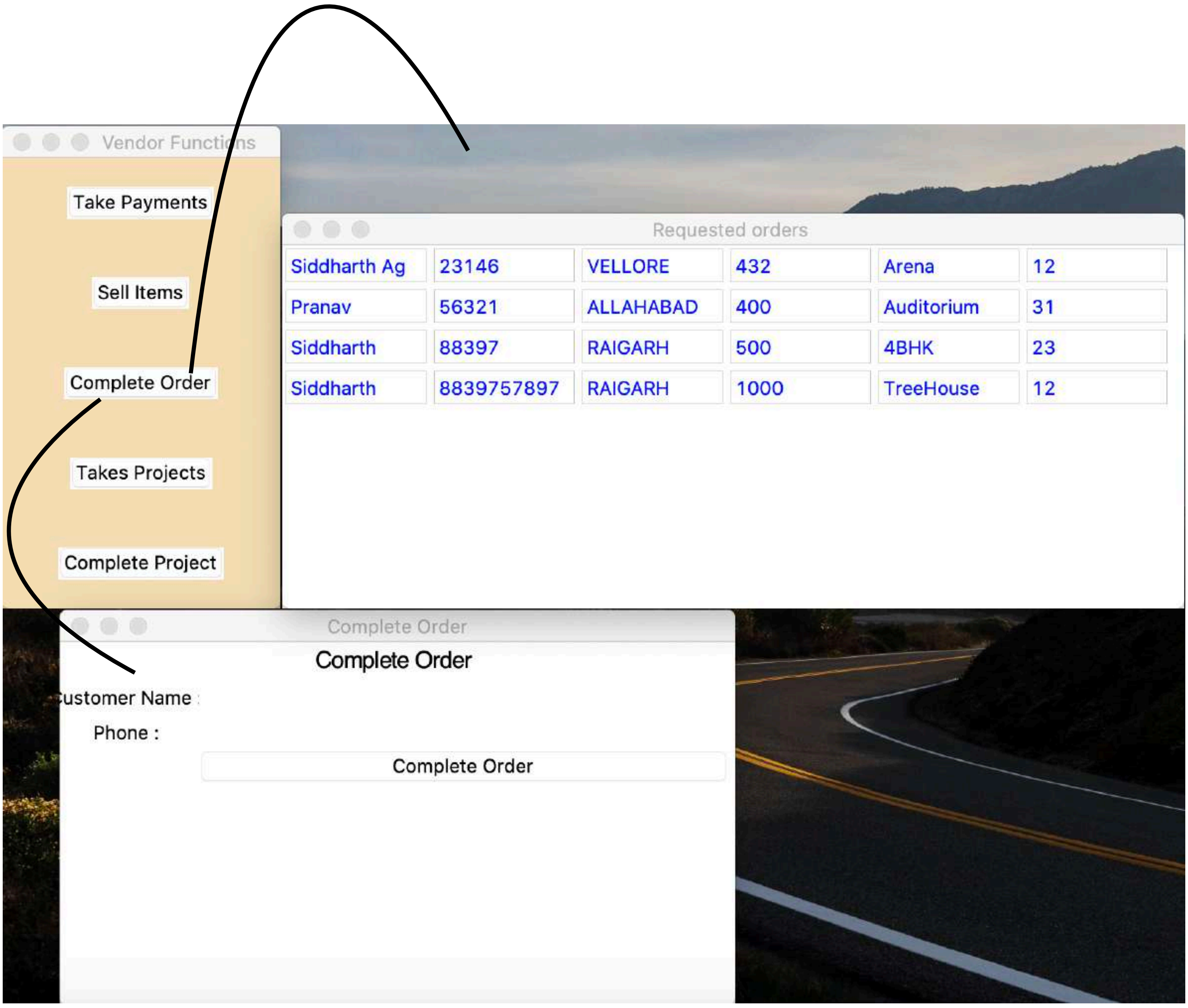
Make Payment

Name :	Siddharth
Phone :	88397
Item :	Hammer



Place Order

:ustomer Name :	Ag
Phone :	23146
Address :	VELLORE
Amount :	432
Project :	Arena
Item :	12



Requested orders

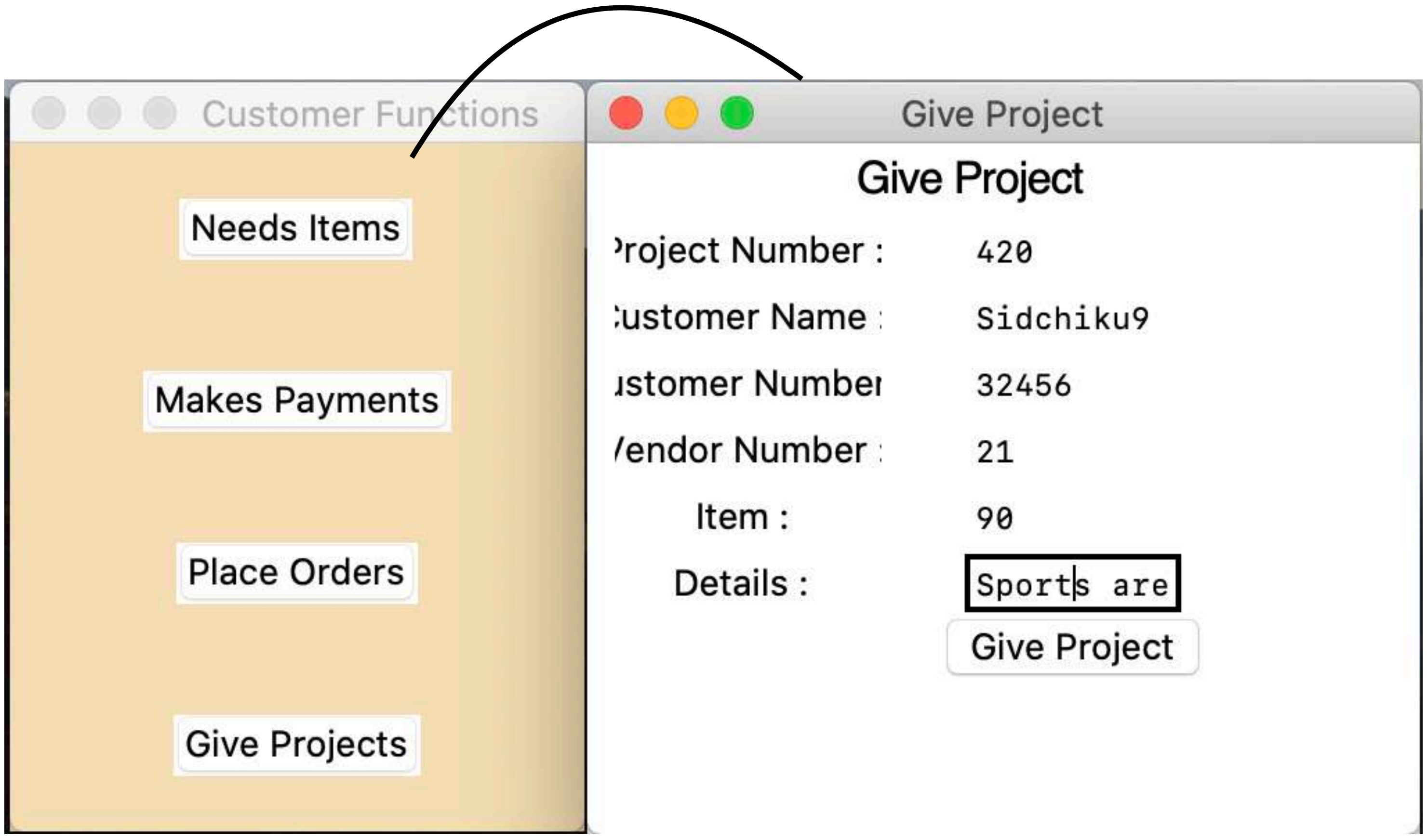
Siddharth Ag	23146	VELLORE	432	Arena	12
Pranav	56321	ALLAHABAD	400	Auditorium	31
Siddharth	88397	RAIGARH	500	4BHK	23
Siddharth	8839757897	RAIGARH	1000	TreeHouse	12

Complete Order

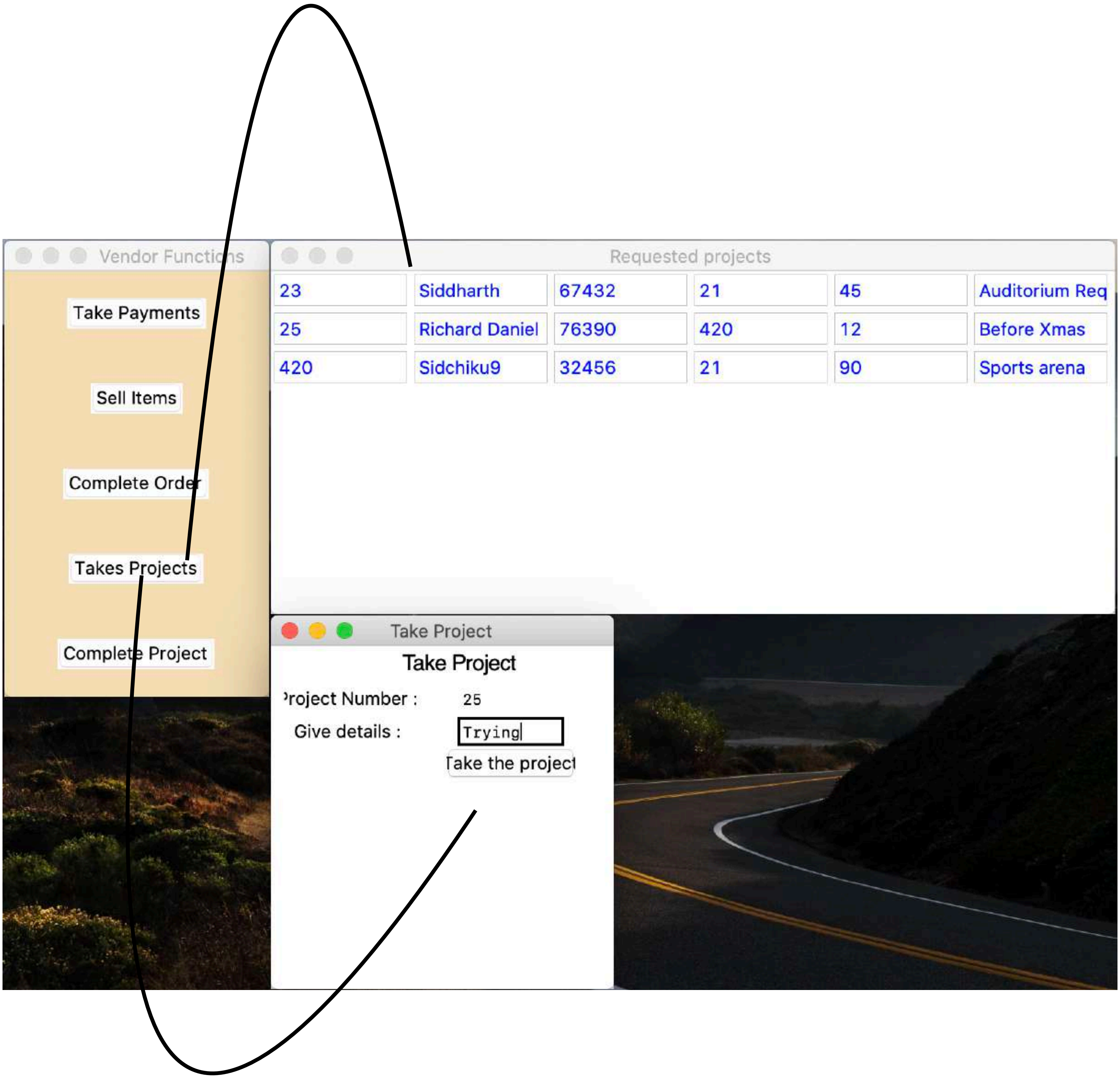
Complete Order

ustomer Name :

Phone :



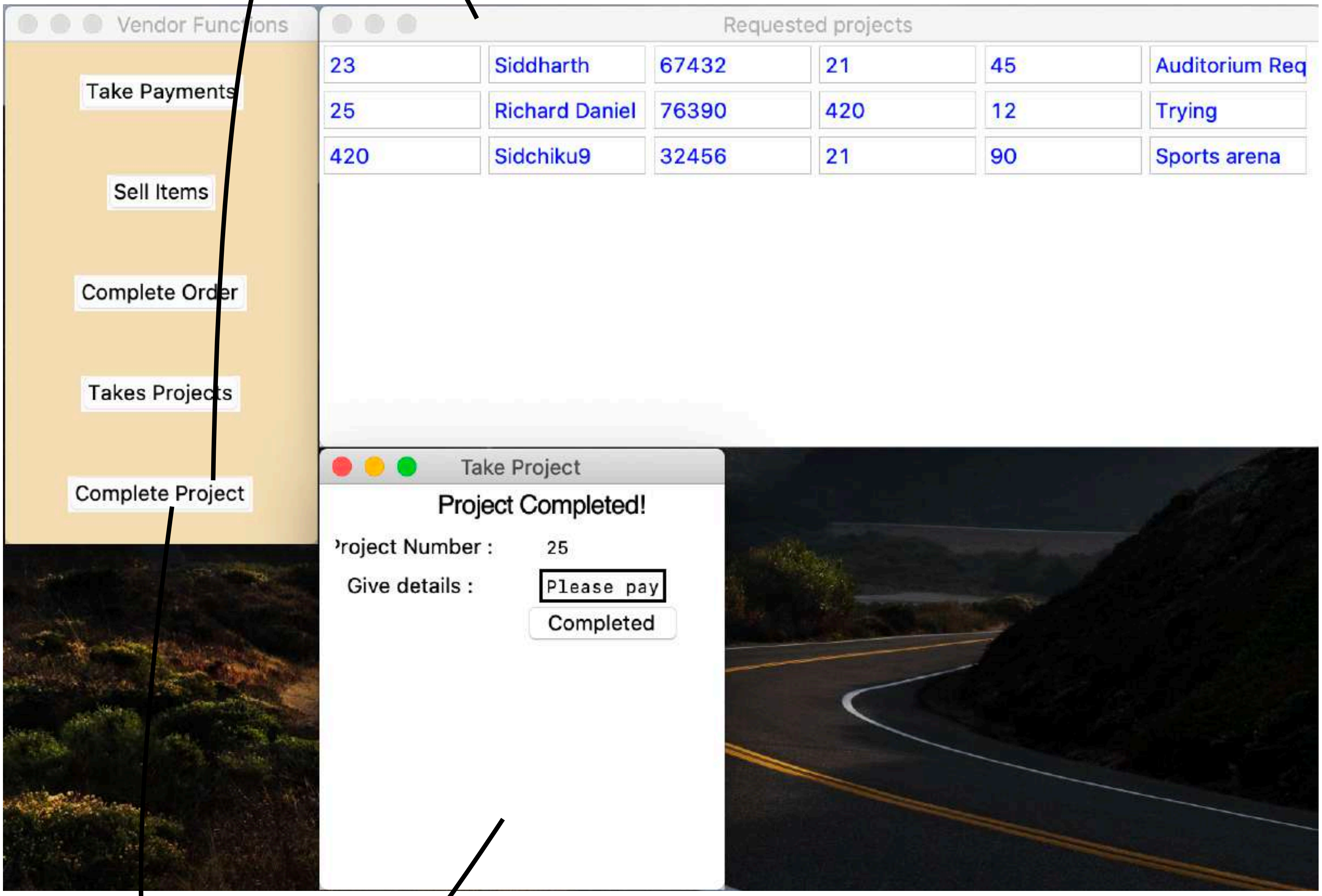
roject Number : 420
ustomer Name : Sidchiku9
ustomer Number 32456
endor Number : 21
Item : 90
Details : Sports are



23	Siddharth	67432	21	45	Auditorium Req
25	Richard Daniel	76390	420	12	Before Xmas
420	Sidchiku9	32456	21	90	Sports arena

Take Project

›roject Number :	25
Give details :	Trying



Requested projects

23	Siddharth	67432	21	45	Auditorium Req
25	Richard Daniel	76390	420	12	Trying
420	Sidchiku9	32456	21	90	Sports arena

Project Completed!

ʼroject Number : 25

Give details : Please pay

Conclusion

With our project we have tried to simplify the management of the construction process. Be it buying items, giving orders or handing out projects, this project has all the necessary functionalities to satisfy your needs. MySQL has been used as the backend with Python and Tkinter as the frontend.

Future Works

The project will be developed so as to be implemented on a network. The python file can be attempted to be launched onto a website using Flask framework which will make the project cross-platform.