

FALL SEMESTER 2018-19
CSE-2004
DATABASE MANAGEMENT SYTEMS
SLOT: D2

FINAL PROJECT REPORT
INSURANCE MANAGEMENT SYSTEM

Submitted by:

Deepak Malpani

17BCE0306

ABSTRACT

The insurance company needs to keep track of details of its target companies, agents, policyholders, their premium payments and the various products that vehicle insurance management system are available with it. Hence it is under tremendous pressure maintaining their day-to-day activities, which is currently being done manually. Entire records have to be updated timely, even a vehicle insurance system slight mistake could complicate things. It is very difficult to handle bulk data since human memory is weaker than electronic counter part.

Hence there is need for an automated system, which can efficiently manage the company, records, provides instant access and one that vehicle insurance management system improves the productivity. As a vehicle insurance system result of this vehicle insurance management system project report automated system, the activities of the company are performed with in the stipulated time and the reliable and efficient service is ensured to its users.

Insurance policy administration system consists of a vehicle insurance system mathematical notation that vehicle insurance management system captures the relationship between policies and objects and the entities that vehicle insurance management system manage policies for those objects.

INTRODUCTION

Relational databases are logical collection of inter-related data in tabular form relational databases have always been core to any management system. Its relevance is profound and hence the need to incorporate new functionalities, utilities becomes important. These are currently the predominant choice in storing financial records manufacturing and logistical information, personnel data and much more.

Relational databases are used in huge management systems like Post Office, Banking, Railway, Defence Logistics. Databases pertaining to Educational Institutions and other large collection of related data.

Relational databases have largely replaced hierarchical databases and network databases because they are easy to understand and use even though they are much less efficient. They have been however challenged by Object Databases and XML databases.

The three leading commercial relational database vendors are Oracle, Microsoft, and IBM. The three leading open Source implementations are MySQL, PostgreSQL, and SQLite.

SYSTEM DESIGN

The software components used in our project are as under:

1. JetBrains PyCharm
2. Oracle 11g database
3. Python 3.5

System requirements:

1. Windows Platform (10 preferred).
2. 4 GB DDR3 RAM.
3. At least 5 GB HDD space FREE.
4. Processor speed 2.0 GHz or greater

OVERVIEW OF THE PROJECT:

SYSTEM DESCRIPTION:

The proposed system is for making easier to manage policy holder details, agent details, policy details, claimant details and payment details. So, this vehicle insurance management system project report will be developed for managing the insurance management system. The overall system is control through the main menu.

The main menu contains 2 parts.

1. Admin Login
2. Agent Login

AGENT LOGIN:

The agent login form links to-

1. Basic agent information like contact details and address which will be shown in customer insurance information window.
2. All the information related to insurances which he has made to his clients.
3. Commission received by him for each insurance made by him respectively.
4. Option to create a vehicle insurance system new policy to any existing/new client.
5. Option to edit the contact information of its client.
6. Option to delete a vehicle insurance system policy of any client in case of policy lapse.

ADMINISTRATOR LOGIN:-

Administrator has rights to-

1. Create new agent
2. Edit agent's information and its commission percentage.
3. Delete an agent's database and all its policies respectively.

TABLE STRUCTURES

AGENT:

| Name | Null? | Type |
|-----------|-------|-----------------------|
| ----- | | |
| AGENT_KEY | | NOT NULL CHAR(5) |
| NAME | | NOT NULL VARCHAR2(50) |
| ADDRESS | | VARCHAR2(70) |
| PHONE | | NOT NULL NUMBER(10) |
| PWD | | NOT NULL VARCHAR2(40) |

CUSTOMER:

| Name | Null? | Type |
|-----------|-------|------------------|
| ----- | | |
| CUSTID | | NOT NULL CHAR(5) |
| NAME | | VARCHAR2(40) |
| MOBILE | | NUMBER |
| ADDRESS | | VARCHAR2(100) |
| AGENT_KEY | | CHAR(5) |

VEHICLE:

| Name | Null? | Type |
|----------|-------|------------------|
| ----- | | |
| VEH_ID | | NOT NULL CHAR(6) |
| CUST_ID | | NOT NULL CHAR(5) |
| VEH_DESC | | VARCHAR2(50) |
| VEH_NUM | | VARCHAR2(12) |
| VEH_TYPE | | VARCHAR2(20) |

CODES

```
from tkinter import *
import os
import cx_Oracle
import random
from tkinter import messagebox
from tkinter import ttk

connectString = os.getenv('db_connect')
con = cx_Oracle.connect('system/deepak123@127.0.0.1/InsuranceManagement')

def stop(root):
    root.destroy()

#Class for inserting new agent
class agent_insert:
    def __init__(self):
        top=self.top=Tk()
        top.geometry("360x360+0+0")
        self.frame=Frame(top,bg='lightgreen',width=360,height=360).pack()
        self.nameins=StringVar()
        self.addrins=StringVar()
        self.passwordins=StringVar()
        self.phoneins=StringVar()
        Label(self.frame, text="New Agent",bg="lightgreen",font=('arial 12')).place(x=140,y=10)
        Label(self.frame, text="Name",bg="lightgreen",font=('arial 10')).place(x=10,y=60)
        ttk.Entry(self.frame,textvariable=self.nameins,font=('arial 10')).place(x=110,y=60)
        Label(self.frame, text="Address",bg="lightgreen",font=('arial 10')).place(x=10,y=110)
        ttk.Entry(self.frame, width=30,textvariable=self.addrins,font=('arial 10')).place(x=110,y=110)
        Label(self.frame, text="Phone no.",bg="lightgreen",font=('arial 10')).place(x=10,y=160)
        ttk.Entry(self.frame,textvariable=self.phoneins,font=('arial 10')).place(x=110,y=160)
        Label(self.frame, text="Password",bg="lightgreen",font=('arial 10')).place(x=10,y=210)
        ttk.Entry(self.frame, show='*',textvariable=self.passwordins,font=('arial 10')).place(x=110,y=210)
        ttk.Button(self.frame, text="Insert", command=self.insert).place(x=70,y=260)
        ttk.Button(self.frame, text="BACK", command=self.admin_page).place(x=170,y=260)
        top.mainloop()
    def insert(self):
        self.agent_key = str(random.randint(10000, 99999))
        a = 'New agent added successfully with agent id = ' + self.agent_key
        cur = con.cursor()
        statement = 'insert into agent (agent_key,name,address,phone,pwd) values(:2,:3,:4,:5,:6)'
```

```

        cur.execute(statement, (self.agent_key, self.nameins.get(), self.addrins.get(), self.phoneins.get(),
self.passwordins.get()))
        messagebox.showinfo("Success", a)
        con.commit()
def stop(self):
    self.top.destroy()
def admin_page(self):
    self.top.destroy()
    Admin_Page()

class agent_login:
def start(self, agent_key):
    top=self.top=Tk()
    self.agent_key=agent_key
    top.geometry("1280x720+0+0")
    self.frame=Frame(top,bg='lightgreen',width=1280,height=720).pack()
    self.custid=StringVar()

    cur=con.cursor()
    statement = "select * from agent where agent_key= '" + agent_key + "' "
    cur.execute(statement)
    arr=cur.fetchall()
    (key,name,address,mobile,password)=arr[0]
    Label(self.frame,text='AGENT DETAILS',bg="lightgreen",font=('arial 12'), fg='Black').place(x=75,y=50)
    Label(self.frame, text='Agent Key',bg="lightgreen", font=('arial 10'), fg='Black').place(x=50, y=100)
    Label(self.frame, text='Name',bg="lightgreen", font=('arial 10'), fg='Black').place(x=50, y=150)
    Label(self.frame, text='Address',bg="lightgreen", font=('arial 10'), fg='Black').place(x=50, y=200)
    Label(self.frame, text='Mobile no.',bg="lightgreen", font=('arial 10'), fg='Black').place(x=50, y=250)

    Label(self.frame, text=agent_key, font=('arial 12'), fg='Black',bg="lightgreen").place(x=150, y=100)
    Label(self.frame, text=name, font=('arial 12'), fg='Black',bg="lightgreen").place(x=150, y=150)
    Label(self.frame, text=address, font=('arial 12'), fg='Black',bg="lightgreen").place(x=150, y=200)
    Label(self.frame,text=mobile, font=('arial 12'), fg='Black',bg="lightgreen").place(x=150,y=250)

    Label(self.frame, text="Enter Customer ID", font=('arial 12'),
fg='Black',bg="lightgreen").place(x=950, y=300)
    ttk.Entry(self.frame, textvariable=self.custid).place(x=1100, y=300)

    ttk.Button(top,text="NEW CUSTOMER",command=self.add_customer).place(x=1050,y=100)
    ttk.Button(top, text="EDIT CUSTOMER", command=self.edit_customer).place(x=1050, y=200)
    ttk.Button(top, text="DELETE CUSTOMER", command=self.delete_customer).place(x=1050, y=350)
    ttk.Button(top,text="LOGOUT",command=self.logout).place(x=640,y=600)

```

```

#Table creation
cur = con.cursor()
statement="select c.custid,c.name,c.mobile,c.address,v.veh_id,v.veh_desc,v.veh_num,v.veh_type
from agent a,customer c, vehicle v where a.agent_key=c.agent_key and c.custid=v.cust_id and
a.agent_key='\" + agent_key + '\" "
cur.execute(statement)
a = cur.fetchall()
con.commit()

```

```

self.treeview = ttk.Treeview(self.frame,height=5)
self.treeview.place(x=50, y=400)
self.treeview.heading('#0', text='Customer ID')

```

```

self.treeview.config(column=('CName', 'CMobile', 'CAddress', 'VId', 'VDesc', 'VNum', 'VType'))
self.treeview.column('#0',width=100)
self.treeview.column('CName',width=100)
self.treeview.column('CMobile', width=100)
self.treeview.column('CAddress', width=200)
self.treeview.column('VId', width=100)
self.treeview.column('VDesc', width=100)
self.treeview.column('VNum', width=100)
self.treeview.column('VType', width=100)

```

```

self.treeview.heading('CName', text='Customer Name')
self.treeview.heading('CMobile', text='Mobile')
self.treeview.heading('CAddress', text='Address')
self.treeview.heading('VId', text='Vehicle ID')
self.treeview.heading('VDesc', text='Vehicle Desc')
self.treeview.heading('VNum', text='Vehicle Number')
self.treeview.heading('VType', text='Vehicle Type')

```

```

if len(a)!=0:
    for i in a:
        (custid, cname, cmobile, cadd, vid,vdesc,vnum,vtype) = i;
        self.treeview.insert("", 'end', custid, text=custid)
        self.treeview.set(custid, 'CName', cname)
        self.treeview.set(custid, 'CMobile', cmobile)
        self.treeview.set(custid, 'CAddress', cadd)
        self.treeview.set(custid, 'VId', vid)
        self.treeview.set(custid, 'VDesc', vdesc)
        self.treeview.set(custid, 'VNum', vnum)
        self.treeview.set(custid, 'VType', vtype)

```

```

self.commission=len(a)*1000 + 200;

```



```

Label(self.frame, text='Commission', font=('arial 12'), fg='Black',bg="lightgreen").place(x=600, y=50)
Label(self.frame,text=str(self.commission), font=('arial 10'),
fg='Black',bg="lightgreen").place(x=625,y=100)
'''

cur = con.cursor()
statement = 'insert into agent_commission (agent_key,commission) values(:2,:3)'
cur.execute(statement, (self.agent_key, self.commission))
'''

top.mainloop()
def stop(self):
    self.treeview.delete(*self.treeview.get_children())
    self.top.destroy()

def edit_customer(self):
    self.top.destroy()
    a=edit_customer()
    a.start(self.agent_key)
def add_customer(self):
    self.stop()
    a=new_customer()
    a.start(self.agent_key)
def delete_customer(self):
    a = 'Record deleted successfully for customer id =' + str(self.custid.get())
    cur = con.cursor()
    statement = "delete from customer where custid= '" + self.custid.get() + "' "
    cur.execute(statement)
    con.commit()
    messagebox.showinfo("Success", a)
def logout(self):
    self.stop()
    login()

```

#NEW-CUSTOMER

```

class new_customer:
    def start(self,agent_key):
        top = self.top = Tk()
        self.agent_key=agent_key
        top.geometry("1280x720+0+0")
        self.frame = Frame(top, bg='lightblue', width=1280, height=720).pack()

        self.name = StringVar()
        self.mobile = StringVar()
        self.address = StringVar()

```

```

self.desc=StringVar()
self.number=StringVar()
self.type=StringVar()

ttk.Label(self.frame, text='CUSTOMER DETAILS').place(x=75, y=50)
Label(self.frame, text='Name').place(x=50, y=150)
Label(self.frame, text='Mobile_no').place(x=50, y=200)
Label(self.frame, text='Address').place(x=50, y=250)

```

```

Entry(self.frame, textvariable=self.name).place(x=155, y=150)
Entry(self.frame, textvariable=self.mobile).place(x=150, y=200)
Entry(self.frame, textvariable=self.address).place(x=150, y=250)

```

```

Label(self.frame, text='VEHICLE DETAILS').place(x=675, y=50)
Label(self.frame, text='Description').place(x=650, y=150)
Label(self.frame, text='Vehicle_no').place(x=650, y=200)
Label(self.frame, text='Type').place(x=650, y=250)

```

```

Entry(self.frame, textvariable=self.desc).place(x=750, y=150)
Entry(self.frame, textvariable=self.number).place(x=750, y=200)
Entry(self.frame, textvariable=self.type).place(x=750, y=250)

```

```

ttk.Button(self.frame, text="Insert", command=self.insert).place(x=375, y=350)
ttk.Button(self.frame, text="Back", command=self.back).place(x=600, y=600)

```

```

def back(self):
    self.top.destroy()
    a=agent_login()
    a.start(self.agent_key)

```

```

def insert(self):
    self.cust_id = str(random.randint(10000, 99999))
    self.veh_id = str(random.randint(100000, 999999))
    a = 'New agent added successfully with customer id = ' + self.cust_id + 'and vehicle id = ' + self.veh_id
    cur = con.cursor()
    statement1 = 'insert into customer (custid,name,mobile,address,agent_key) values(:2,:3,:4,:5,:6)'
    statement2 = 'insert into vehicle (veh_id,cust_id,veh_desc,veh_num,veh_type) values(:2,:3,:4,:5,:6)'
    cur.execute(statement1, (self.cust_id, self.name.get(), self.mobile.get(), self.address.get(),
self.agent_key))
    cur.execute(statement2, (self.veh_id, self.cust_id, self.desc.get(), self.number.get(), self.type.get()))
    messagebox.showinfo("Success", a)
    con.commit()

```

```

class edit_customer:
    def start(self, agent_key):
        top=self.top=Tk()
        self.agent_key=agent_key
        top.geometry("1280x720+0+0")
        self.frame = Frame(top, bg='lightblue', width=1280, height=720).pack()

        self.custid=StringVar()
        self.name = StringVar()
        self.mobile = StringVar()
        self.address = StringVar()
        self.desc = StringVar()
        self.number = StringVar()
        self.type = StringVar()

        style=ttk.Style()
        style.configure("BW.TLabel",foreground="Black",background="#41acf4")

        ttk.Label(self.frame, text="EDIT AGENT DETAILS", font=('arial 15'),style="BW.TLabel").place(x=75,
y=50)
        Label(self.frame, text="Enter Customer ID", font=('arial 10'), fg='Black', bg='#41acf4').place(x=450,
y=150)
        ttk.Entry(self.frame, textvariable=self.custid).place(x=650, y=150)
        ttk.Entry(self.frame, textvariable=self.name).place(x=75, y=250)
        ttk.Button(self.frame, text="Update Name", command=self.edit_name).place(x=225, y=250)
        ttk.Entry(self.frame, textvariable=self.mobile).place(x=75, y=350)
        ttk.Button(self.frame, text="Update Mobile no.",command=self.edit_mobile).place(x=225, y=350)
        ttk.Entry(self.frame, textvariable=self.address).place(x=75, y=450)
        ttk.Button(self.frame, text="Update Address",command=self.edit_address).place(x=225, y=450)

        Label(self.frame, text="EDIT VEHICLE DETAILS", font=('arial 15'), fg='Black',
bg='#41acf4').place(x=875, y=50)

        ttk.Entry(self.frame, textvariable=self.desc).place(x=775, y=250)
        ttk.Button(self.frame, text="Update Description", command=self.edit_desc).place(x=925, y=250)
        ttk.Entry(self.frame, textvariable=self.number).place(x=775, y=350)
        ttk.Button(self.frame, text="Update Vehicle no.",command=self.edit_vehno).place(x=925, y=350)
        ttk.Entry(self.frame, textvariable=self.type).place(x=775, y=450)
        ttk.Button(self.frame, text="Update Vehicle type",command=self.edit_type).place(x=925, y=450)
        ttk.Button(self.frame, text="Back", command=self.back).place(x=600, y=600)

        top.mainloop()

```

```

def back(self):
    self.top.destroy()
    a=agent_login()
    a.start(self.agent_key)

def edit_name(self):
    a = 'Info edited successfully'
    cur = con.cursor()
    statement = "UPDATE customer SET NAME=:1 WHERE custid=:2"
    cur.execute(statement, (self.name.get(), self.custid.get()))
    messagebox.showinfo("Success", a)
    con.commit()

def edit_mobile(self):
    a = 'Info edited successfully'
    cur = con.cursor()
    statement = "UPDATE customer SET MOBILE=:1 WHERE custid=:2"
    cur.execute(statement, (self.mobile.get(), self.custid.get()))
    messagebox.showinfo("Success", a)
    con.commit()

def edit_address(self):
    a = 'Info edited successfully'
    cur = con.cursor()
    statement = "UPDATE customer SET ADDRESS=:1 WHERE custid=:2"
    cur.execute(statement, (self.address.get(), self.custid.get()))
    messagebox.showinfo("Success", a)
    con.commit()

def edit_desc(self):
    a = 'Info edited successfully'
    cur = con.cursor()
    statement = "UPDATE vehicle SET VEH_DESC=:1 WHERE cust_id=:2"
    cur.execute(statement, (self.desc.get(), self.custid.get()))
    messagebox.showinfo("Success", a)
    con.commit()

def edit_vehno(self):
    a = 'Info edited successfully'
    cur = con.cursor()
    statement = "UPDATE vehicle SET VEH_num=:1 WHERE cust_id=:2"
    cur.execute(statement, (self.number.get(), self.custid.get()))
    messagebox.showinfo("Success", a)
    con.commit()

def edit_type(self):
    a = 'Info edited successfully'

```

```

cur = con.cursor()
statement = "UPDATE vehicle SET VEH_TYPE=:1 WHERE cust_id=:2"
cur.execute(statement, (self.type.get(), self.custid.get()))
messagebox.showinfo("Success", a)
con.commit()

```

#Class for new admin page

```
class Admin_Page:
```

```
    def __init__(self):
```

```
        top=self.top=Tk()
```

```
        top.geometry("1280x720+0+0")
```

```
        top.resizable(False,False)
```

```
        self.left=Frame(top,width=800,height=720,bg="#4298f4").pack(side=LEFT)
```

```
        self.right = Frame(top, width=480, height=720, bg="#4298f4").pack(side=LEFT)
```

```
        self.agent_key=StringVar()
```

```
        self.name=StringVar()
```

```
        self.phone=StringVar()
```

```
        self.password=StringVar()
```

```
        self.address=StringVar()
```

```
        self.agent_key_edit=StringVar()
```

```
        Label(self.left,text="ADMINISTRATOR LOGIN",font=('arial 30
bold'),fg='Black',bg='#4298f4').place(x=0,y=0)
```

```
        ttk.Button(self.left,text="NEW AGENT",command=self.new_agent).place(x=80,y=80)
```

```
        ttk.Button(self.left,text="DELETE AGENT",command=self.delete).place(x=80,y=150)
```

```
        ttk.Entry(self.left,textvariable=self.agent_key).place(x=200,y=155)
```

```
        Label(self.right, text="EDIT AGENT DETAILS", font=('arial 15'), fg='Black',
bg='#41acf4').place(x=800,y=20)
```

```
        Label(self.right, text="Enter Agent ID", font=('arial 10'), fg='Black', bg='#41acf4').place(x=1000,
y=100)
```

```
        ttk.Entry(self.right, textvariable=self.agent_key_edit).place(x=990, y=135)
```

```
        ttk.Entry(self.right, textvariable=self.name).place(x=850, y=175)
```

```
        ttk.Button(self.right,text="Update Name",command=self.edit_name).place(x=1000,y=173)
```

```
        ttk.Entry(self.right, textvariable=self.address).place(x=850, y=225)
```

```
        ttk.Button(self.right, text="Update Address").place(x=1000, y=223)
```

```
        ttk.Entry(self.right, textvariable=self.phone).place(x=850, y=275)
```

```
        ttk.Button(self.right, text="Update Phone No.").place(x=1000, y=273)
```

```
        ttk.Entry(self.right, textvariable=self.password).place(x=850, y=325)
```

```

ttk.Button(self.right, text="Update Password").place(x=1000, y=323)

ttk.Button(self.right, text="LOGOUT", command=self.logout).place(x=80, y=250)

#Table Creation
cur = con.cursor()
cur.execute('SELECT * FROM AGENT')
a = cur.fetchall()
con.commit()

self.treeview = ttk.Treeview(self.right)
self.treeview.place(x=20, y=400)
self.treeview.heading('#0', text='Agent ID')
self.treeview.config(columnn=('Name', 'Address', 'Phone', 'Password'))
self.treeview.heading('Name', text='Name')
self.treeview.heading('Address', text='Address')
self.treeview.heading('Phone', text='Phone')
self.treeview.heading('Password', text='Password')

for i in a:
    (key, name, add, no, pwd) = i;
    self.treeview.insert("", 'end', key, text=key)
    self.treeview.set(key, 'Name', name)
    self.treeview.set(key, 'Address', add)
    self.treeview.set(key, 'Phone', no)
    self.treeview.set(key, 'Password', pwd)

top.mainloop()
def new_agent(self):
    self.stop()
    agent_insert()
def logout(self):
    self.stop()
    login()

def delete(self):
    a = 'Record deleted successfully with agent id =' + str(self.agent_key.get())
    cur = con.cursor()
    statement = "delete from agent where agent_key= " + self.agent_key.get() + " "
    cur.execute(statement)
    con.commit()
    messagebox.showinfo("Success", a)
def stop(self):
    self.top.destroy()

```

```

def edit_name(self):
    a = 'Info edited successfully'
    cur = con.cursor()
    statement = "UPDATE agent SET NAME=:1 WHERE AGENT_KEY=:2"
    cur.execute(statement,(self.name.get(),self.agent_key_edit.get()))
    #cur.execute(statement)
    messagebox.showinfo("Success", a)
    con.commit()

```

```

class login:

```

```

    def __init__(self):
        top=self.top=Tk()
        top.title('LOGIN')
        top.geometry('480x360+0+0')
        top.resizable(False,False)

```

```

        self.left=Frame(top,width=240,height=360,bg="lightpink").pack(side=LEFT)
        self.right=Frame(top,width=240,height=360,bg="lightblue").pack(side=RIGHT)

```

```

        Label(self.left,text="ADMIN LOGIN",font=('arial 13'),bg='lightpink').place(x=50,y=0)
        Label(self.left, text="Admin ID", bg='lightpink').place(x=20, y=60)
        Label(self.left, text="Password", bg='lightpink').place(x=20, y=120)

```

```

        self.adminID=StringVar()
        self.adminpwd=StringVar()

```

```

        ttk.Entry(self.left,textvariable=self.adminID,width=15).place(x=125,y=60)
        ttk.Entry(self.left, textvariable=self.adminpwd,show='*', width=15).place(x=125, y=120)

```

```

        ttk.Button(self.left,text="LOGIN",command=self.admin_login).place(x=100,y=180)

```

```

        Label(self.right, text="AGENT LOGIN", font=('arial 13'), bg='lightblue').place(x=290, y=0)
        Label(self.right, text="Agent ID", bg='lightblue').place(x=260, y=60)
        Label(self.right, text="Password", bg='lightblue').place(x=260, y=120)

```

```

        ttk.Button(self.right, text="LOGIN", command=self.agent_login).place(x=325, y=180)

```

```

self.agent_key=StringVar()
self.agentpwd=StringVar()
ttk.Entry(self.right, textvariable=self.agent_key, width=15).place(x=360, y=60)
ttk.Entry(self.right, show='*', textvariable=self.agentpwd, width=15).place(x=360, y=120)
top.mainloop()
def admin_login(self):
    if self.adminID.get()=='admin' and self.adminpwd.get()=='123456':
        self.stop()
        Admin_Page()
    else:
        messagebox.showerror('Error', 'Invalid Credentials')
def stop(self):
    self.top.destroy()

def agent_login(self):
    con = cx_Oracle.connect('system/deepak123@127.0.0.1/InsuranceManagement')
    cur = con.cursor()
    statement = "select * from agent where agent_key=:1 and pwd=:2 "
    cur.execute(statement, (self.agent_key.get(), self.agentpwd.get()))
    a = cur.fetchall()
    if len(a)==0:
        messagebox.showerror('Error', 'Enter valid login credentials')
    else:
        self.stop()
        a=agent_login()
        a.start(self.agent_key.get())

class start:
    def __init__(self):
        root =self.root= Tk()
        root.title('LOGIN')
        root.geometry('1200x628+0+0')
        root.resizable(False, False)
        C = Canvas(root, bg="blue", height=250, width=300)
        filename = PhotoImage(file="C:\\Users\\admin\\Desktop\\vehicle.png")
        background_label = Label(root, image=filename)
        background_label.place(x=0, y=0, relwidth=1, relheight=1)

        Button(root, text='TAKE ME TO LOGIN PAGE', background="lightblue", font=('arial
14'), command=self.login).place(x=480, y=450)
        root.mainloop()
    def login(self):
        self.root.destroy()

```


login()

start()

SNAPSHOTS

HOME SCREEN:



LOGIN:

The screenshot shows the login page of the 'VEHICLE INSURANCE MANAGEMENT SYSTEM'. The page is divided into two main sections: 'ADMIN LOGIN' (pink background) and 'AGENT LOGIN' (blue background). Each section contains a 'Login ID' field, a 'Password' field, and a 'LOGIN' button. The window title bar at the top reads 'LOGIN'.

| ADMIN LOGIN | | AGENT LOGIN | |
|--------------------------------------|--------------------------|--------------------------------------|--------------------------|
| Login ID | <input type="text"/> | Agent ID | <input type="text"/> |
| Password | <input type="password"/> | Password | <input type="password"/> |
| <input type="button" value="LOGIN"/> | | <input type="button" value="LOGIN"/> | |

ADMIN PAGE:

ADMINISTRATOR LOGIN

NEW AGENT

DELETE AGENT

LOGOUT

EDIT AGENT DETAILS

Enter Agent ID

Update Name

Update Address

Update Phone No.

Update Password

| Agent ID | Name | Address | Phone | Password |
|----------|-----------|-----------|------------|----------|
| 68588 | Sam | 29, Japur | 9630758147 | 123456 |
| 41523 | Nick | ZYZ | 9630758341 | 123456 |
| 99879 | Debabrata | TJ, ZYZ | 7124536980 | 456789 |

NEW AGENT:

tk

New Agent

Name

Address

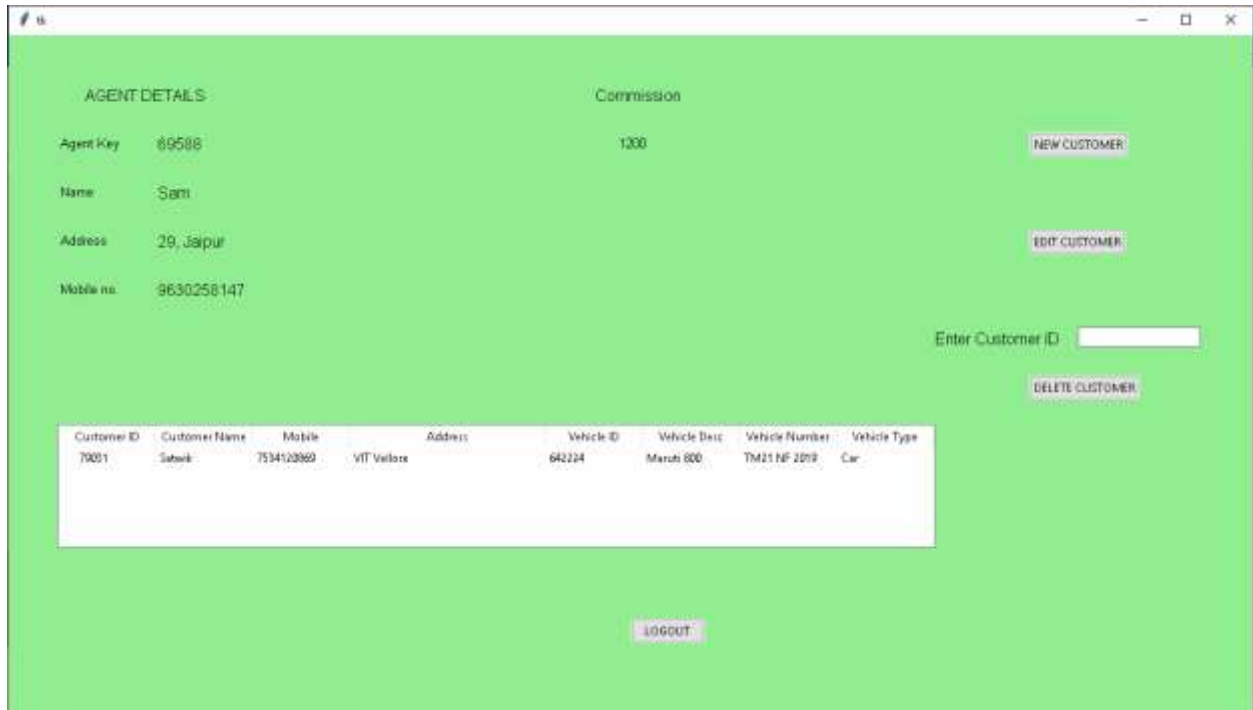
Phone no.

Password

Insert

BACK

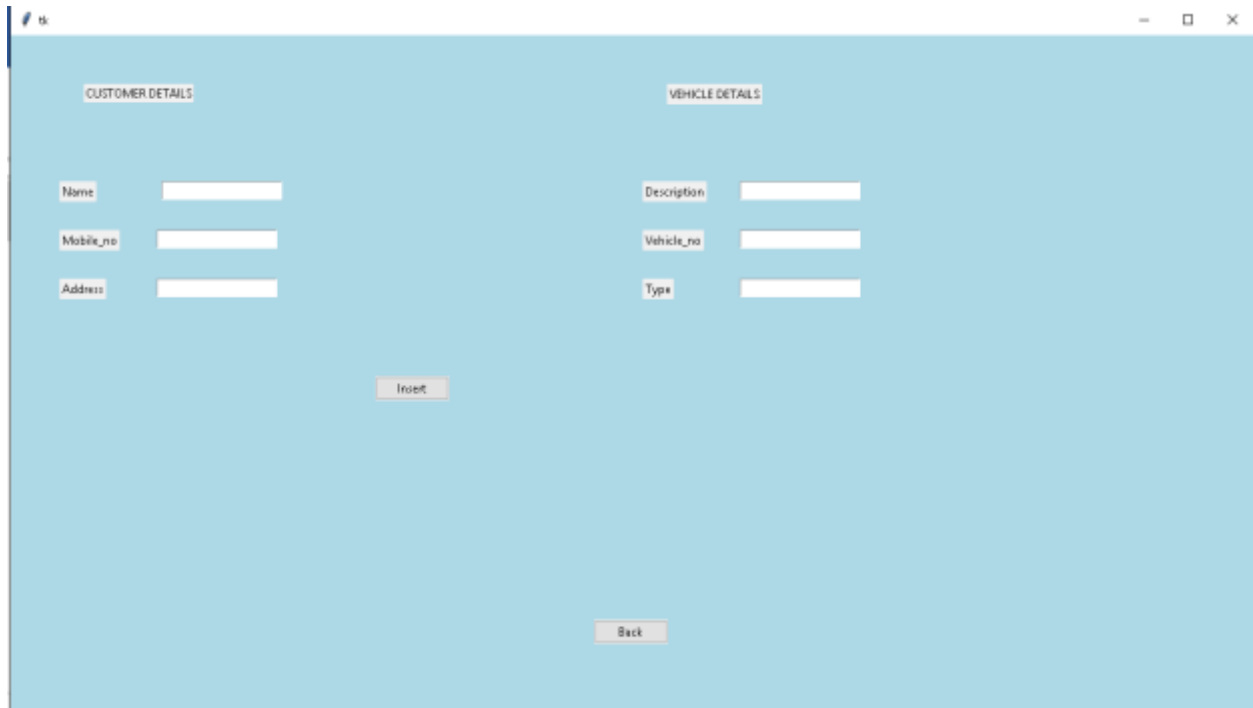
AGENT LOGIN:



The Agent Login form is displayed on a light green background. It features two main sections: 'AGENT DETAILS' and 'Commission'. The 'AGENT DETAILS' section includes fields for Agent Key (89588), Name (Sami), Address (29, Jaipur), and Mobile no. (9630258147). The 'Commission' section shows a value of 1200. To the right of these fields are three buttons: 'NEW CUSTOMER', 'EDIT CUSTOMER', and 'DELETE CUSTOMER'. Below these buttons is a text input field labeled 'Enter Customer ID'. At the bottom center is a 'LOGOUT' button. A table is also present, showing customer and vehicle information.

| Customer ID | Customer Name | Mobile | Address | Vehicle ID | Vehicle Desc | Vehicle Number | Vehicle Type |
|-------------|---------------|------------|--------------|------------|--------------|----------------|--------------|
| 7001 | Sateek | 7534123899 | VIT Varanasi | 642224 | Maruti 800 | TM21 NF 2019 | Car |

NEW CUSTOMER:



The New Customer form is displayed on a light blue background. It is divided into two sections: 'CUSTOMER DETAILS' and 'VEHICLE DETAILS'. The 'CUSTOMER DETAILS' section includes input fields for Name, Mobile_no, and Address. The 'VEHICLE DETAILS' section includes input fields for Description, Vehicle_no, and Type. Below the 'CUSTOMER DETAILS' section is an 'Insert' button. At the bottom center is a 'Back' button.

EDIT CUSTOMER:

EDIT AGENT DETAILS

EDIT VEHICLE DETAILS

Enter Customer ID

Update Name

Update Mobile no.

Update Address

Update Description

Update Vehicle no.

Update Vehicle type

Back

CONCLUSION

A VEHICLE INSURANCE SYSTEM computerized insurance management system has been developed and the system was tested with sample data.

The system results in regular timely preparations of required outputs. In comparison with manual system the benefits under a vehicle insurance system computer system are considerable in the saving of manpower working hours and Effort.

Provision for addition , updating and deletion of customers is there in the system .It is observed that vehicle insurance management system proper filing system has been adopted for future reference . The entire project runs on windows environments.

The system can be used to make better management described at appropriate time. The user gets amount and timely information system.