Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Computer Engineering

Academic Year: 2019-20 **Semester: IV** **Division: OSTL Lab**

A Mini Project Report on

[PATIENT MANAGEMENT SYSTEM]

Roll No	Name
7	Rakesh Choudhary
12	Satyam Gupta
13	Yogesh Hole

Mini Project Title	Front End (08)	Back End (04)	Viva (05)	Documentation (03)	Total Marks (20)	Teacher Signature with date

Table of contents

Contents

Chapter 1

Introduction

Overview

This report discusses the result of the work done on '**Patient Management System**' on Python IDLE Platform .The report provides common platform for facilitating the use of technological approach developed by our Health Department and integration of various tools developed during the execution of the project.

Objective

The goal of the project is to:

1) Understand the working of software developed, with use of various tools already defined in Python.
2) Implement basic knowledge of **Database Connectivity** in this software application.

Methodology

To implement the above goals, the following methodology needs to be followed:

1) Specifying the application and various components of GUI.
2) Specifying the bindings between the tasks and resources either manually or by the design tools.
3) Development of Python Program by using IDLE 3.7.
4) Storing data values in database and extracting them for further use.
5) Use of my SQL lite for database system

Chapter 2

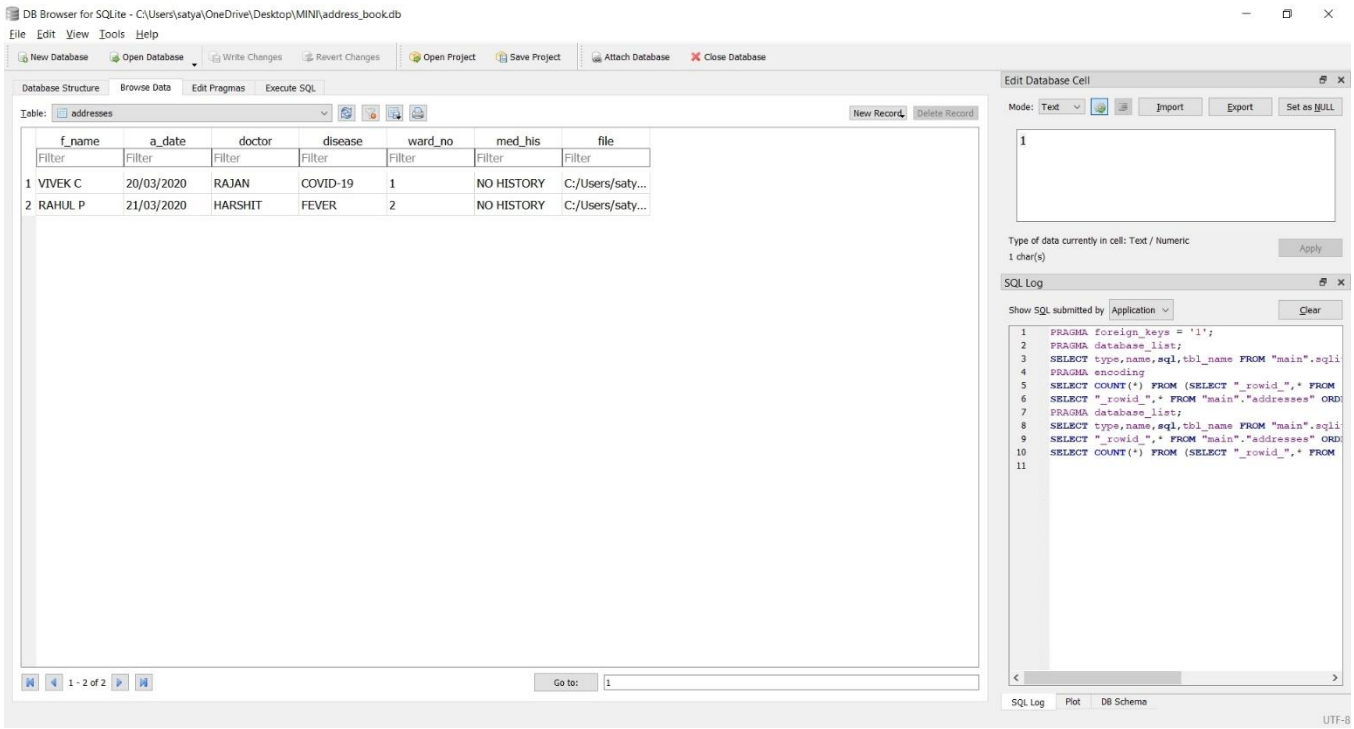
Problem definition

We are going to build a software based Patient management system which will upload the data of patient that will have several features:

1) Record of Patient Detail.
2) Database Management System (DBMS).
3) Data stored and extraction using SQL lite .
4) Data Entry work.
5) GUI (Graphical User Interface).

Chapter 3

Database



Chapter 4

Source code

```
from tkinter import *

from tkinter import filedialog

import tkinter as tk

from tkinter.messagebox import askyesno

import sqlite3


top = Tk()

top.title("Patient Management System")

top.geometry("400x400")

top.config(bg='yellow')




def destroy():

        top.destroy()

def upload(event):

   print("ready to upload")
```

```
    top = Tk()
    top.title("upload details")
    top.geometry("400x400")
    top.config(bg='green')
    def upload_photo(event):

        top.filename =  filedialog.askopenfilename(initialdir = "/",title = "Select file",filetypes = (("jpeg files",".jpg"),("all files",".*")))

        print(top.filename)
    def submit():
            conn = sqlite3.connect('address_book.db')
            c=conn.cursor()
            c.execute("INSERT INTO addresses VALUES (:f_name,:a_date,:doctor,:disease,:ward_no,:med_his,:file)",
              {
                  'f_name' :e1.get(),
                  'a_date' :e2.get(),
                  'doctor' :e3.get(),
                  'disease':e4.get(),
                  'ward_no':e5.get(),
                  'med_his':e6.get(),
                  'file':top.filename

              })
```

```
        conn.commit()
        conn.close()

def clear1():
    e1.delete(first=0,last=22)
    e2.delete(first=0,last=22)
    e3.delete(first=0,last=22)
    e4.delete(first=0,last=22)
    e5.delete(first=0,last=22)
    e6.delete(first=0,last=22)

def destroy():
        top.destroy()

photo1 = Button(top, text = "upload photo",activebackground = "pink", activeforeground = "blue")
photo1.pack()
photo1.bind('<Button-1>',upload_photo)
#creating label
uname = Label(top, text = "Full name").place(x = 30,y = 50)

#creating label
Admit_date = Label(top, text = "Admit date").place(x = 30, y = 90)
Doctor = Label(top, text = "Doctor").place(x = 30, y = 130)
disease  = Label(top, text = "disease").place(x = 30, y = 170)
```

```
    ward_no = Label(top, text = "ward no.").place(x = 30, y = 210)

    medical_history  = Label(top, text = "Medical history").place(x = 30, y = 250)



    e1 = Entry(top,width=20)

    e2 = Entry(top,  width = 20)

    e3 = Entry(top, width = 20)

    e4 = Entry(top, width = 20)

    e5 = Entry(top, width = 20)

    e6 = Entry(top, width = 20)

    e1.place(x=130,y=50)

    e2.place(x=130,y=90)

    e3.place(x=130,y=130)

    e4.place(x=130,y=170)

    e5.place(x=130,y=210)

    e6.place(x=130,y=250)

    sbmitbtn =
Button(top,text="Submit",activebackground="pink",activeforeground="blue",command=submit).place(
x=110,y=300)


    clear = Button(top, text = "clear",activebackground = "yellow", activeforeground =
"blue",command=clear1).place(x=190,y=300)

    quit1 = Button(top, text = "Quit",activebackground = "yellow", activeforeground =
"blue",command=destroy).place(x=260,y=300)
```

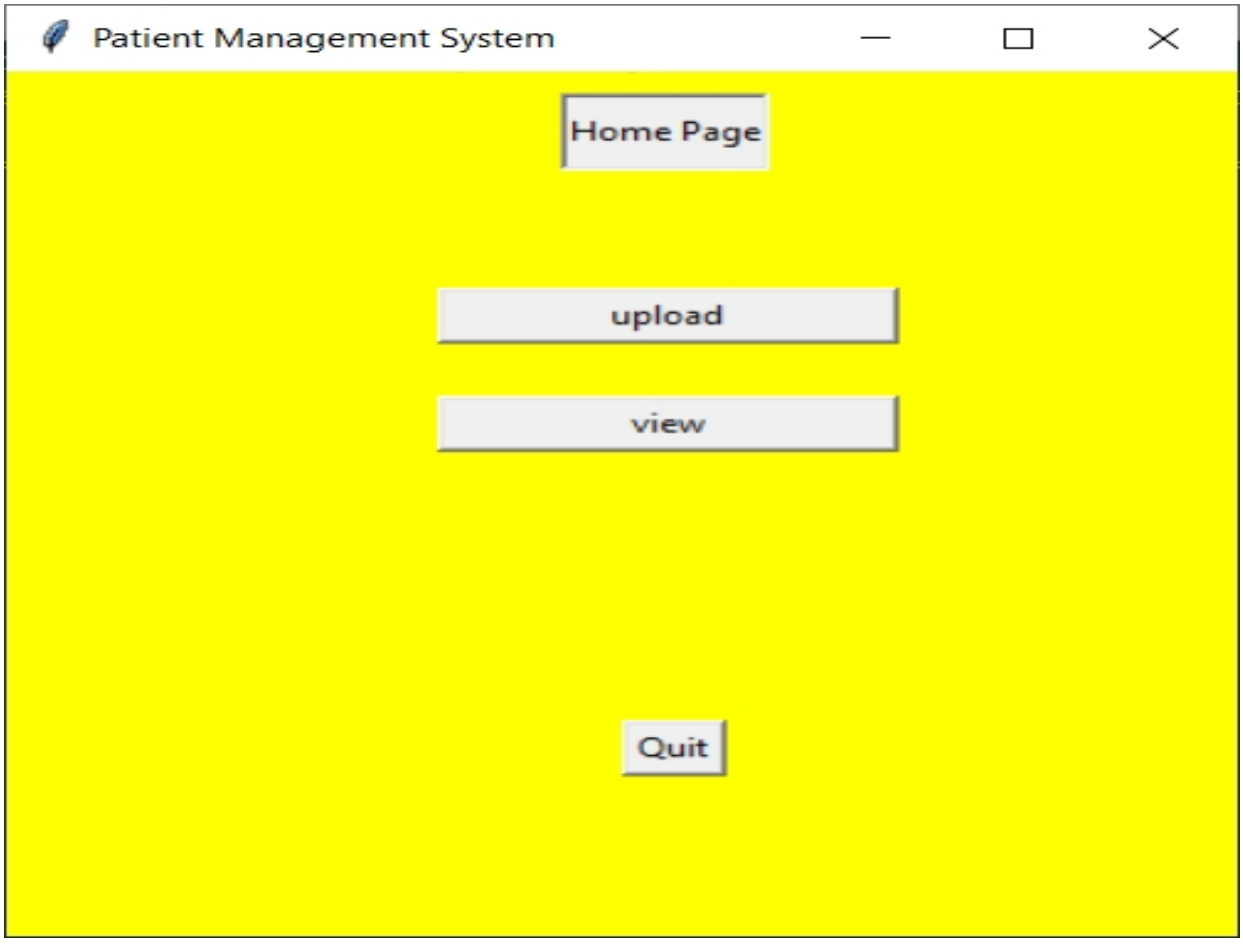
```
    top.mainloop()




def view(event):
    top = Tk()

    top.geometry("400x400")
    print("ready to view")

    top.title('view details')
    query_label="
    q1 = Entry(top,width=20)
    q1.place(x=120,y=10)

    def query():
        conn = sqlite3.connect('address_book.db')
        c=conn.cursor()

        c.execute("SELECT *, oid FROM addresses")
        records = c.fetchall()
```

```
        print_records =''
        n=int (q1.get()) -1
        for record in records[n]:
            print_records += str(record) + "\n" + "\n"



        query_label=Label(top,text=print_records,width=20)
        query_label.place(x=120,y=65)



        conn.commit()
        conn.close()
def clean():
        q1.delete(first=0,last=22)
        top.destroy()
reg_no = Label(top, text = "Patient no.").place(x = 10, y = 10)
query_btn = Button(top,text='show records',command=query).place(x=250,y=10)
uname = Label(top, text = "Full name").place(x = 30,y = 65)

#creating label
Admit_date = Label(top, text = "Admit date").place(x = 30, y = 95)
Doctor = Label(top, text = "Doctor").place(x = 30, y = 125)
disease  = Label(top, text = "disease").place(x = 30, y = 155)
```

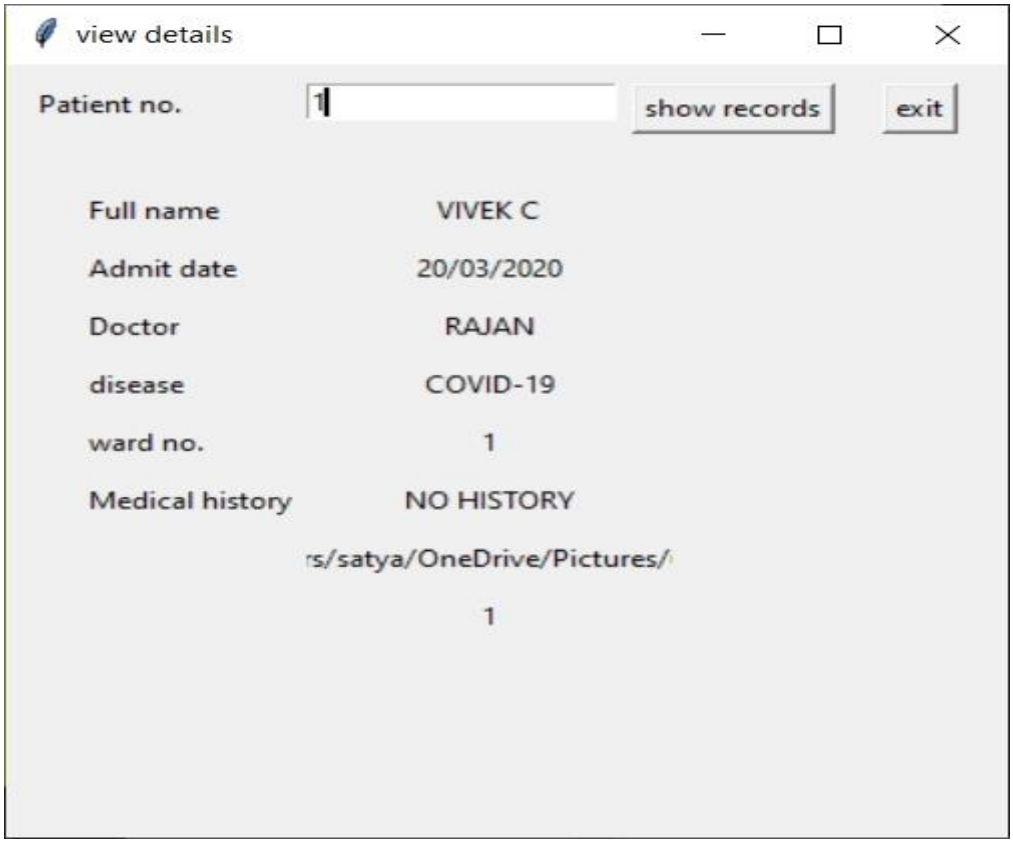
```
    ward_no = Label(top, text = "ward no.").place(x = 30, y = 185)
    medical_history  = Label(top, text = "Medical history").place(x = 30, y = 215)

    clear = Button(top,text='exit',command=clean).place(x=350,y=10)
    top.mainloop()

uname = Label(top, text = "Home Page",relief=SUNKEN,height=2).place(x=180,y=10)
widget = Button(None,text='upload',width=20,highlightcolor="blue")
widget1 = Button(None,text='view',width=20)
widget.place(x=140,y=100)
widget1.place(x=140,y=150)

widget.bind('<Button-1>', upload)
widget1.bind('<Button-1>',view)
quit1 = Button(top, text = "Quit",activebackground = "yellow", activeforeground =
"blue",command=destroy).place(x=200,y=300)

widget.mainloop()
widget1.mainloop()
top.mainloop()
```

Chapter 5

Results




upload details
upload photo
Full name
VIVEK C
Admit date
20/03/2020
Doctor
RAJAN
disease
COVID-19
ward no.
1
Medical history
NO HISTORY
Submit
clear
Quit


view details
Patient no.
1
show records
exit
Full name
VIVEK C
Admit date
20/03/2020
Doctor
RAJAN
disease
COVID-19
ward no.
1
Medical history
NO HISTORY
rs/satya/OneDrive/Pictures/
1

Chapter 6

Conclusion

The Patient management system is developed using Python IDLE Platform fully meets the objectives of the system which it has been developed .The system has reached a steady state where all bugs have been eliminated. The system is operated at high level of efficiency and all the Doctors and Patient associated with the system understands its advantage. The system solves the problem. It was intended to solve as requirement specification.

Chapter 7

Reference

For execution of '**Patient Management System**' following references have been used:

1) Designing GUI : https://www.tutorialspoint.com/python/python_gui_programming.ht
2) Database connectivity : https://www.w3schools.com/python/python_mysql_getstarted.asp
3) Reference book : Java- Learning Python