A Project report on

COMPUTERIZED LIBRARY MANAGEMENT SYSTEM

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology
In
Computer Science and Engineering

Submitted by

B.Swathi	19H51A0531
D.Sahithi	19H51A0535
J.Shyam Kumar	19H51A0542

Under the esteemed guidance of

Dr.K.L.S Soujanya

Professor and HOD Dept.of IT

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC & JNTUH, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

2019 - 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini Project-1 report entitled **"COMPUTERIZED LIBRARY MANAGEMENT SYSTEM"** being submitted by **B. Swathi** ***(19H51A0531),*** **D. Sahithi** ***(19H51A0535),*** **J. Shyam Kumar** ***(19H51A0542),*** in partial fulfillment for the award of **Bachelor of Technology in** Computer Science and Engineering is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Dr. KLS Soujanya	**Dr. K. Vijaya Kumar**
Professor and HOD	**Professor and HOD**
Dept. of IT	**Dept. of CSE**

Submitted for viva voice Examination held on ___________

External Examiner

ABSTRACT

Manual process of keeping student records, book records, account details, managing is very difficult. To eliminate this manual system, Computerized Library management system is the project which aims in developing a computerized system to maintain all computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new users etc. This project has many features like It has a facility where users after logging in their account can see list of books, issue book and can search books too. To store all the information in the database from where user will place their query and get the results on the basis of their query. Only valid users will be able to access this system. Through this system it will be easy to manage various details of library along with the records, issue, search, delete, return of the books. Overall this project of ours is being developed to help the users as well as staff of the library to maintain the library in the best way possible and also reduce the human effort.

TABLE OF CONTENTS

1. INTRODUCTION

Managing a library requires knowledge of library management and skills to perform the activities. It involves planning, decision making, organizing, collecting information, controlling and monitoring the various functions. Library plays an important role in all schools and colleges. It is an important part of every school and college, it helps the librarian to keep records of available books as well as issued books. The purpose of a library management system is to operate a library with efficiency and at reduced costs. The activities of book cataloging, indexing, circulation recording and checking are done by the software. Such software eliminates the need for repetitive manual work and minimizes the chances of errors. The library management system software helps in reducing operational costs. Managing a library manually is labor intensive and an immense amount of paperwork is involved. An automated system reduces the need for manpower and stationery. This leads to lower operational costs. Checking and verification of books in the library can be done within a few hours. The automated system saves a considerable amount of time as opposed to the manual system

1.1 SCOPE

- This application can be used by any library to automate the process of manually maintaining the records. Where students can check the available books in the library.

1.2 OBJECTIVES

- The software keeps track of all the information about the books and their complete details.
- Save cost.
- After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost.

2. BACKGROUND WORK

2.1 EXISTING SYSTEMS:

Manual operating system is vulnerable to human errors. For instance, a librarian who miss fills borrows records or indexes a book incorrectly slows down the process and wastes employee’s time. Manual system are unable to store large amount of data efficiently, with manual systems staff spends a lot of time on mechanical, clerical tasks rather than liaising with library visitors. On a simple level, locating a precise book within the local library system is time consuming without a linked computer network. With a manual operating system librarian relay on regular contact with their members and generally communicate by mail.

2.2 GAPS IN EXISTING SOLUTION

- It takes more effort and physical space to keep the track of information and to keep details secure.
- When mistakes or corrections are needed, often a manual transactions must be completely redone rather than just updated.
- Inconsistency in data entry, room for error, miss keying information.
- Reduction in sharing information and customer service.
- .Time consuming and costly to produce reports.
- Lack of security.
- Duplication of data entry.

3. PROPOSED SYSTEM

3.1 OVERALL SYSTEM DESIGN & DESCRIPTION

The proposed Computerized Library Management System has a profound effect on both work of library staff and the amount of information available to library patrons. The library management system is a software to manage function of a library. In addition, it allows streamlined management of fine details of books such as author name, tittle name, and many other important details. So, it is easier to search for books and find the right material foe users and the librarian. This is essential to track the information like, who has borrowed any material, etc. The system is developed and designed with an aim to facilitate efficient management to the users to manage a modern library with accurate data management.

3.1.1 ADVANTAGES OF PROPOSED SYSTEM

- Simple and easy to operate**.**
- Not easy for data loss.
- Increase librarian efficiencies.
- Need short time to find any user information.
- Search, add, update, and view library materials.
- Help to manage library functions constructively.
- Reduce library’s operating cost.
- Remove manual process to issue books and maintain records.

3.2 MODULES DESCRIPTION

- Library Registration:

  The first procedure is the registration of the people who arrive to the library. The registration contains fields like enter the name, id, branch, department, password, date of joining, salary.
- Login registration:

  The administrator has provide the authority to login directly after the registration. The login fields contains name, id, password, and role.

- Add books:

  Admin has the authority to add, issue, delete, view, search of book available to/from the system. The software keeps track of all the information about the books in the library.
- Delete books:

  Admin has the authority to delete a book.
- Issue books:

  Admin has the authority to issue books to the students. Issue book contain fields like employee id, student id, book id.
- Search book:

  Admin and the students has the authority to search the books.
- View book list:

  Admin and students has the authority to view the books.

3.3 PYTHON TKINTER

TKinter is a python binding to the TK GUI toolkit. It is the standard python interface to the TK toolkit, and is python's standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X install of python.

3.4 DESIGN A GRAPHICAL INTERFACE USING TKINTER LIBRARY

Graphical User Interface (GUI) is nothing but a desktop application which helps in the desktop interact with the computer.

- GUI apps like Text-Editors are used to create, read, update and delete different types of files.
- GUI apps like Sudoku, Chess and Solitaire are games which you can play.
- GUI apps like Google Chrome, Firefox and Microsoft Edge are used to browse through the Internet.
- IDE's for coding are on a GUI app.

3.5 REQUIREMENT SPECIFICATION

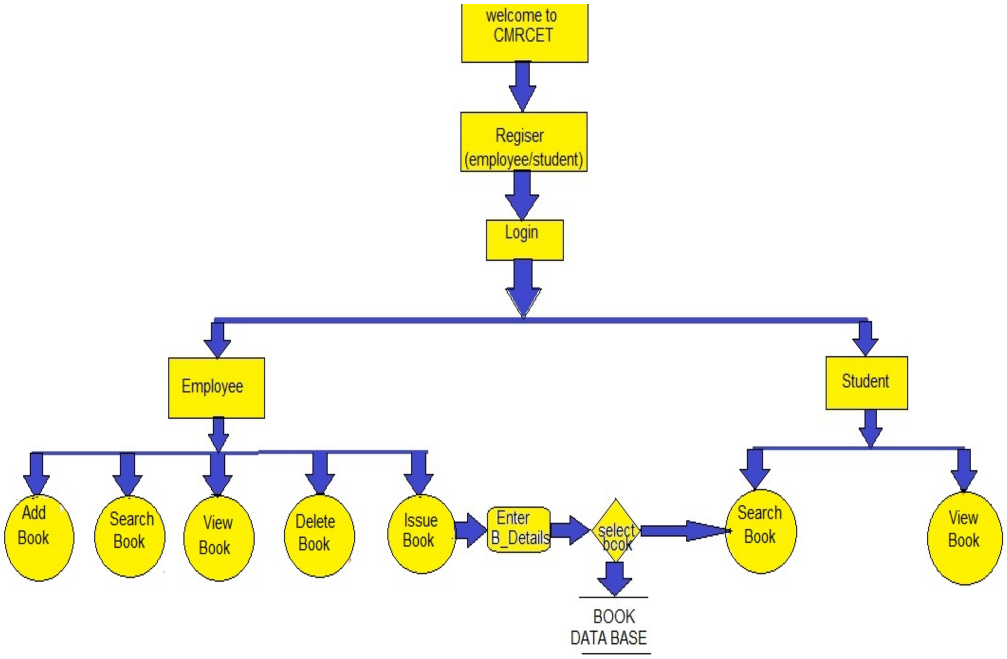
3.5.1 HARDWARE REQUIREMENTS

- Processor : Intel core i3
- Hard disk : 10GB
- Laptop : Dell

3.5.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 10
- Database : MYSQL
- Languages : Python

3.6 PROPOSED SYSTEM ARCHITECTURE



4. DESIGNING UML DIAGRAM

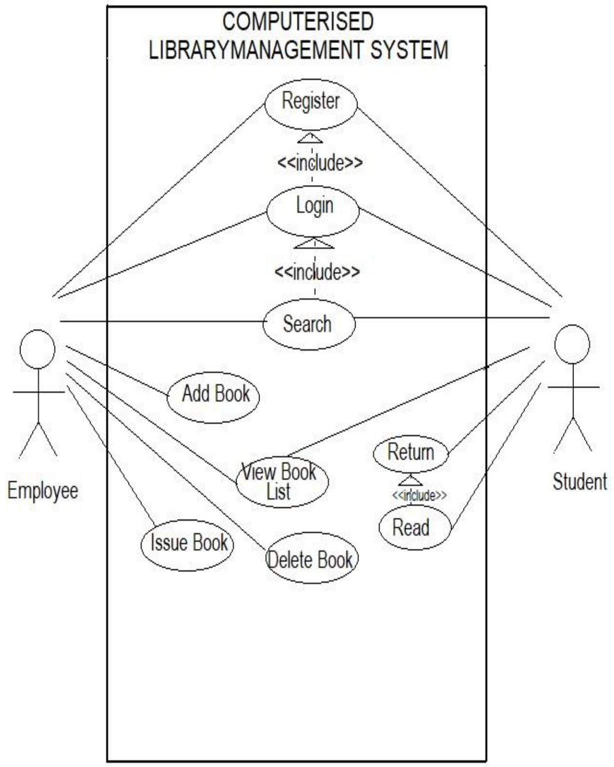


Fig: 4.1 Use Case Diagram

Use case diagrams referred as a Behavior model or diagram. It simply describes and displays the relation or interaction between the employee or student .It describes different actions that a system performs in collaboration to achieve something with one or more users of the system.

5. RESULTS & DISCUSSION

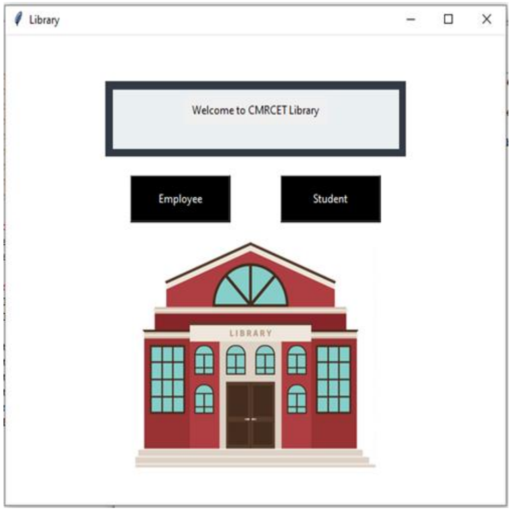
Main code

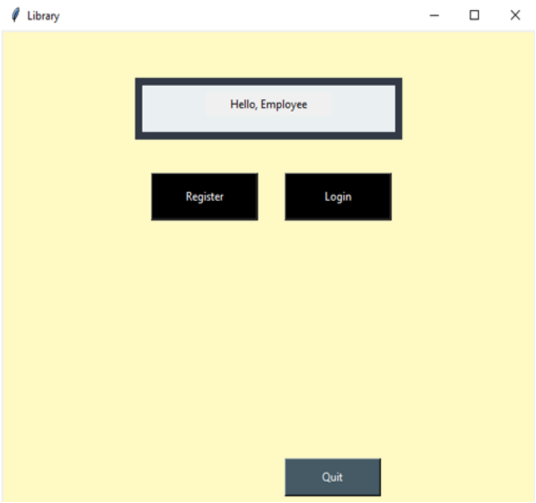
```
from tkinter import *
from PIL import ImageTk,Image
import pymysql
from tkinter import messagebox
from AddBooks import *
from DeleteBook import *
from ViewBooks import *
from SearchBook import *
from IssueBook import *

# Add your own database name and password here to reflect in the code
mypass = "root"
mydatabase="db"

# Enter Table Names here
empTable = "empdetail" #Employee Table
stuTable = "studetail" #Student Table

root = Tk()
root.title("Library")
root.minsize(width=400,height=400)
root.geometry("600x500")
count = 0
empFrameCount = 0

con =
pymysql.connect(host="localhost",user="root",password=mypass,database=mydat
abase)
cur = con.cursor()


'''
This are the menus after logging in
'''
def empMenu():

    global
headingFrame1,headingFrame2,headingLabel,SubmitBtn,Canvas1,labelFrame,ba
ckBtn
    headingFrame1.destroy()
    headingFrame2.destroy()
    headingLabel.destroy()
    Canvas1.destroy()
    SubmitBtn.destroy()
    Canvas1 = Canvas(root)

    Canvas1.config(bg="#f7f1e3",width = newImageSizeWidth, height =
newImageSizeHeight)
    Canvas1.pack(expand=True,fill=BOTH)

    headingFrame1 = Frame(root,bg="#333945",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingFrame2 = Frame(headingFrame1,bg="#EAF0F1")
    headingFrame2.place(relx=0.01,rely=0.05,relwidth=0.98,relheight=0.9)

    headingLabel = Label(headingFrame2, text="Employee MENU", fg='black')
    headingLabel.place(relx=0.25,rely=0.15, relwidth=0.5, relheight=0.5)

    btn1 = Button(root,text="Add Book Details",bg='black', fg='white',
command=addBooks)
    btn1.place(relx=0.28,rely=0.3, relwidth=0.45,relheight=0.1)

    btn2 = Button(root,text="Delete Book",bg='black', fg='white', command=delete)
    btn2.place(relx=0.28,rely=0.4, relwidth=0.45,relheight=0.1)

    btn3 = Button(root,text="View Book List",bg='black', fg='white', command=View)
    btn3.place(relx=0.28,rely=0.5, relwidth=0.45,relheight=0.1)

    btn4 = Button(root,text="Search Book",bg='black', fg='white',
command=searchBook)
    btn4.place(relx=0.28,rely=0.6, relwidth=0.45,relheight=0.1)

    btn5 = Button(root,text="Issue Book to Student",bg='black', fg='white', command
= issueBook)
    btn5.place(relx=0.28,rely=0.7, relwidth=0.45,relheight=0.1)

    backBtn = Button(root,text="<  BACK",bg='#455A64', fg='white',
command=Employee)
    backBtn.place(relx=0.5,rely=0.9, relwidth=0.18,relheight=0.08)

def stuMenu():

    global
headingFrame1,headingFrame2,headingLabel,SubmitBtn,Canvas1,btn1,btn2,btn3
,btn4,btn5,backBtn
    headingFrame1.destroy()
    headingFrame2.destroy()
    headingLabel.destroy()
    Canvas1.destroy()
    SubmitBtn.destroy()
    Canvas1 = Canvas(root)

    Canvas1.config(bg="#dff9fb",width = newImageSizeWidth, height = 
newImageSizeHeight)
    Canvas1.pack(expand=True,fill=BOTH)

    headingFrame1 = Frame(root,bg="#333945",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingFrame2 = Frame(headingFrame1,bg="#EAF0F1")
    headingFrame2.place(relx=0.01,rely=0.05,relwidth=0.98,relheight=0.9)

    headingLabel = Label(headingFrame2, text="Student MENU", fg='black')
    headingLabel.place(relx=0.25,rely=0.15, relwidth=0.5, relheight=0.5)

    btn1 = Button(root,text="View Book List",bg='black', fg='white',command=View)
    btn1.place(relx=0.28,rely=0.35, relwidth=0.45,relheight=0.1)

    btn2 = Button(root,text="Search Book",bg='black', 
fg='white',command=searchBook)
    btn2.place(relx=0.28,rely=0.45, relwidth=0.45,relheight=0.1)

    backBtn = Button(root,text="<  BACK",bg='#455A64', fg='white', 
command=Student)
    backBtn.place(relx=0.5,rely=0.9, relwidth=0.18,relheight=0.08)


'''
This Section handles the database
'''
def gettingEmpDetails():

    EmpId = en1.get()
    name = en2.get()
    password = en3.get()
    dept = en4.get()
    doj = en5.get()
    sal = en6.get()

    try:
        if (type(int(EmpId)) == int):
            pass
        else:
            messagebox.showinfo("Invalid Value","Employee ID should be an integer")
            return
    except:
        messagebox.showinfo("Invalid Value","Employee ID should be an integer")
        return
    try:
        if (type(float(sal)) == float or type(int(sal)) == int):
            pass
        else:
            messagebox.showinfo("Invalid Value","Salary should be a float/int value")
            return
    except:
        messagebox.showinfo("Invalid Value","Salary should be a float/int value")
        return

    sql = "insert into "+empTable+" values
('"+EmpId+"','"+name+"','"+password+"','"+dept+"','"+doj+"','"+sal+"')"
    try:
        cur.execute(sql)
        con.commit()
    except:
        messagebox.showinfo("Error inserting","Cannot add data to Database")

    print(EmpId)
    print(name)
    print(password)
    print(dept)
    print(doj)
    print(sal)
    en1.delete(0, END)
    en2.delete(0, END)
    en3.delete(0, END)
    en4.delete(0, END)
    en5.delete(0, END)
    en6.delete(0, END)

def gettingStuDetails():

    Rollno = en1.get()
    name = en2.get()
    password = en3.get()
    dept = en4.get()
    sem = en5.get()
    batch = en6.get()

    try:
        if (type(int(Rollno)) == int):
            pass
        else:
            messagebox.showinfo("Invalid Value","Roll number should be an integer")
            return
    except:
        messagebox.showinfo("Invalid Value","Roll number should be an integer")
        return
    sql = "insert into "+stuTable+" values
('"+Rollno+"','"+name+"','"+password+"','"+dept+"','"+sem+"','"+batch+"')"
    try:
        cur.execute(sql)
        con.commit()
    except:
        messagebox.showinfo("Error inserting","Cannot add data to Database")

    print(Rollno)
    print(name)
    print(password)
    print(dept)
    print(sem)
    print(batch)
    en1.delete(0, END)
    en2.delete(0, END)
    en3.delete(0, END)
    en4.delete(0, END)
    en5.delete(0, END)
    en6.delete(0, END)

def gettingLoginDetails():

    login = en1.get()
    name = en2.get()
    password = en3.get()
    role = en4.get()
    role.lower()

    if (role == 'emp'):
        sqlLoginID = "select empid from "+empTable+" where password =
'"+password+"'"
        sqlName = "select name from "+empTable+" where password =
'"+password+"'"

        try:
            cur.execute(sqlLoginID)
            for i in cur:
                getLoginID = i[0]
            cur.execute(sqlName)
            for i in cur:
                getName = i[0]

            if(getLoginID == login and getName == name):
                empMenu()
                messagebox.showinfo("SUCCESS","You have successfully logged in")
            else:
                messagebox.showerror("Failure","Can't log in, check your credentials")
        except:
            messagebox.showinfo("FAILED","Please check your credentials")
    elif (role == 'stu'):
        sqlLoginID = "select rollno from "+stuTable+" where password = 
'"+password+"'"
        sqlName = "select name from "+stuTable+" where password = 
'"+password+"'"

        try:
            cur.execute(sqlLoginID)
            for i in cur:
                getLoginID = i[0]
            cur.execute(sqlName)
            for i in cur:
                getName = i[0]

            if(getLoginID == login and getName == name):
                stuMenu()
                messagebox.showinfo("SUCCESS","You have successfully logged in")
            else:
                messagebox.showerror("Failure","Can't log in, check your credentials")
        except:
            messagebox.showinfo("FAILED","Please check your credentials")
    else:
        messagebox.showinfo("EXCEPTION","Role can only be emp or stu")
        return

    print(login)
    print(name)
    print(password)
    print(role)
    en1.delete(0, END)
    en2.delete(0, END)
    en3.delete(0, END)
    en4.delete(0, END)

def EmpRegister():

    global labelFrame

    global count
    count += 1

    if(count>=2):
        labelFrame.destroy()

    global en1,en2,en3,en4,en5,en6
    labelFrame = Frame(root,bg='#044F67')
    labelFrame.place(relx=0.2,rely=0.44,relwidth=0.6,relheight=0.42)

    # Employee ID
    lb1 = Label(labelFrame,text="Emp ID : ", bg='#044F67', fg='white')
    lb1.place(relx=0.05,rely=0.05)

    en1 = Entry(labelFrame)
    en1.place(relx=0.3,rely=0.05, relwidth=0.62)

    #Employee Name
    lb2 = Label(labelFrame,text="Name : ", bg='#044F67', fg='white')
    lb2.place(relx=0.05,rely=0.2)

    en2 = Entry(labelFrame)
    en2.place(relx=0.3,rely=0.2, relwidth=0.62)

    #Employee Paswword
    lb3 = Label(labelFrame,text="Password : ", bg='#044F67', fg='white')
    lb3.place(relx=0.05,rely=0.35)

    en3 = Entry(labelFrame)
    en3.place(relx=0.3,rely=0.35, relwidth=0.62)

    #Employee Department
    lb4 = Label(labelFrame,text="Department : ", bg='#044F67', fg='white')
    lb4.place(relx=0.05,rely=0.5)

    en4 = Entry(labelFrame)
    en4.place(relx=0.3,rely=0.5, relwidth=0.62)

    #Employee Date of Joining
    lb5 = Label(labelFrame,text="DOJ : ", bg='#044F67', fg='white')
    lb5.place(relx=0.05,rely=0.65)

    en5 = Entry(labelFrame)
    en5.place(relx=0.3,rely=0.65, relwidth=0.62)

    # Employee Salary
    lb5 = Label(labelFrame,text="Salary : ", bg='#044F67', fg='white')
    lb5.place(relx=0.05,rely=0.8)

    en6 = Entry(labelFrame)
    en6.place(relx=0.3,rely=0.8, relwidth=0.62)

    #Submit Button
    SubmitBtn = Button(root,text="SUBMIT",bg='#264348',
fg='white',command=gettingEmpDetails)
    SubmitBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)
```

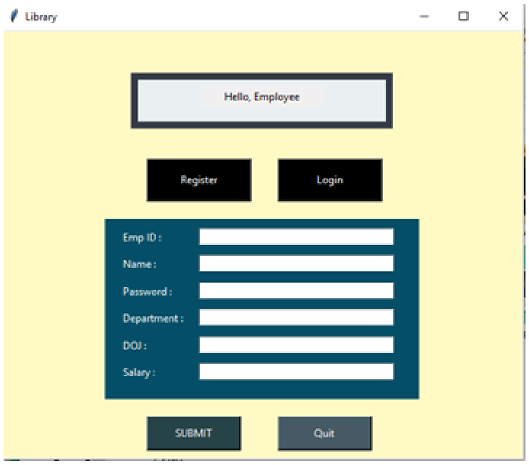
```
# Login both for Employee and Student
def Login():

    global labelFrame

    global count
    count += 1

    if(count>=2):
        labelFrame.destroy()

    global en1,en2,en3,en4,SubmitBtn,btn1,btn2

    labelFrame = Frame(root,bg='#044F67')
    labelFrame.place(relx=0.2,rely=0.44,relwidth=0.6,relheight=0.3)

    # Login ID
    lb1 = Label(labelFrame,text="Login ID : ", bg='#044F67', fg='white')
    lb1.place(relx=0.05,rely=0.1)

    en1 = Entry(labelFrame)
    en1.place(relx=0.3,rely=0.1, relwidth=0.62)

    # Name
    lb2 = Label(labelFrame,text="Name : ", bg='#044F67', fg='white')
    lb2.place(relx=0.05,rely=0.3)

    en2 = Entry(labelFrame)
    en2.place(relx=0.3,rely=0.3, relwidth=0.62)

    # Paswword
    lb3 = Label(labelFrame,text="Password : ", bg='#044F67', fg='white')
    lb3.place(relx=0.05,rely=0.5)

    en3 = Entry(labelFrame)
    en3.place(relx=0.3,rely=0.5, relwidth=0.62)

    # Role
    lb4 = Label(labelFrame,text="Role : ", bg='#044F67', fg='white')
    lb4.place(relx=0.05,rely=0.7)

    en4 = Entry(labelFrame)
    en4.place(relx=0.3,rely=0.7, relwidth=0.62)

    #Submit Button
    SubmitBtn = Button(root,text="SUBMIT",bg='#264348',
fg='white',command=gettingLoginDetails)
    SubmitBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)
```

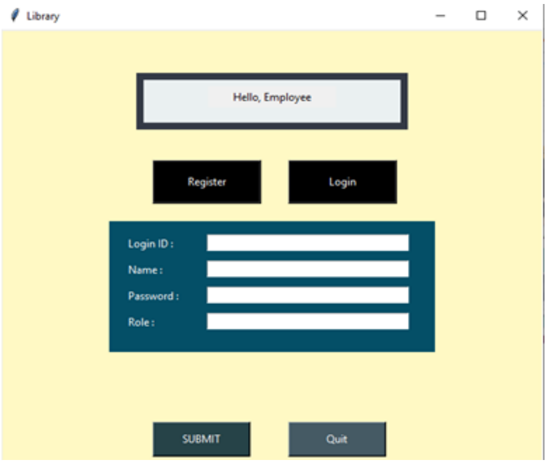
```
# Student Registration
def studentRegister():

    global labelFrame

    global count
    count += 1

    if(count>=2):
        labelFrame.destroy()

    global en1,en2,en3,en4,en5,en6

    labelFrame = Frame(root,bg='#044F67')
    labelFrame.place(relx=0.2,rely=0.44,relwidth=0.6,relheight=0.42)

    # Student Roll no
    lb1 = Label(labelFrame,text="Roll No : ", bg='#044F67', fg='white')
    lb1.place(relx=0.05,rely=0.05)

    en1 = Entry(labelFrame)
    en1.place(relx=0.3,rely=0.05, relwidth=0.62)

    # Sudent Name
    lb2 = Label(labelFrame,text="Name : ", bg='#044F67', fg='white')
    lb2.place(relx=0.05,rely=0.2)

    en2 = Entry(labelFrame)
    en2.place(relx=0.3,rely=0.2, relwidth=0.62)

    # Student Password
    lb3 = Label(labelFrame,text="Password : ", bg='#044F67', fg='white')
    lb3.place(relx=0.05,rely=0.35)

    en3 = Entry(labelFrame)
    en3.place(relx=0.3,rely=0.35, relwidth=0.62)

    # Student Department
    lb4 = Label(labelFrame,text="Dept : ", bg='#044F67', fg='white')
    lb4.place(relx=0.05,rely=0.5)

    en4 = Entry(labelFrame)
    en4.place(relx=0.3,rely=0.5, relwidth=0.62)

     # Student Semester
    lb5 = Label(labelFrame,text="Semester : ", bg='#044F67', fg='white')
    lb5.place(relx=0.05,rely=0.65)

    en5 = Entry(labelFrame)
    en5.place(relx=0.3,rely=0.65, relwidth=0.62)

    # Student Batch
    lb6 = Label(labelFrame,text="Batch : ", bg='#044F67', fg='white')
    lb6.place(relx=0.05,rely=0.8)

    en6 = Entry(labelFrame)
    en6.place(relx=0.3,rely=0.8, relwidth=0.62)

    #Submit Button
    SubmitBtn = Button(root,text="SUBMIT",bg='#264348',
fg='white',command=gettingStuDetails)
    SubmitBtn.place(relx=0.28,rely=0.9, relwidth=0.18,relheight=0.08)

# Employee Home Page
def Employee():

    global headingFrame1,headingFrame2,headingLabel,btn1,btn2,Canvas1
    headingFrame1.destroy()
    headingFrame2.destroy()
    headingLabel.destroy()
    Canvas1.destroy()
    btn1.destroy()
    btn2.destroy()

    Canvas1 = Canvas(root)

    Canvas1.config(bg="#FFF9C4",width = newImageSizeWidth, height =
newImageSizeHeight)
    Canvas1.pack(expand=True,fill=BOTH)


    headingFrame1 = Frame(root,bg="#333945",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingFrame2 = Frame(headingFrame1,bg="#EAF0F1")
    headingFrame2.place(relx=0.01,rely=0.05,relwidth=0.98,relheight=0.9)

    headingLabel = Label(headingFrame2, text="Hello, Employee", fg='black')
    headingLabel.place(relx=0.25,rely=0.15, relwidth=0.5, relheight=0.5)

    btn1 = Button(root,text="Register",bg='black',
fg='white',command=EmpRegister)
    btn1.place(relx=0.28,rely=0.3, relwidth=0.2,relheight=0.1)

    btn2 = Button(root,text="Login",bg='black', fg='white', command=Login)
    btn2.place(relx=0.53,rely=0.3, relwidth=0.2,relheight=0.1)

    btn3 = Button(root,text="Quit",bg='#455A64', fg='white', command=root.quit)
    btn3.place(relx=0.53,rely=0.9, relwidth=0.18,relheight=0.08)

# Student Home Page
def Student():

    global headingFrame1,headingFrame2,headingLabel,btn1,btn2,Canvas1
    headingFrame1.destroy()
    headingFrame2.destroy()
    headingLabel.destroy()
    Canvas1.destroy()
    btn1.destroy()
    btn2.destroy()

    Canvas1 = Canvas(root)

    Canvas1.config(bg="#FFF9C4",width = newImageSizeWidth, height =
newImageSizeHeight)
    Canvas1.pack(expand=True,fill=BOTH)


    headingFrame1 = Frame(root,bg="#333945",bd=5)
    headingFrame1.place(relx=0.25,rely=0.1,relwidth=0.5,relheight=0.13)

    headingFrame2 = Frame(headingFrame1,bg="#EAF0F1")
    headingFrame2.place(relx=0.01,rely=0.05,relwidth=0.98,relheight=0.9)

    headingLabel = Label(headingFrame2, text="Hello, Student", fg='black')
    headingLabel.place(relx=0.25,rely=0.15, relwidth=0.5, relheight=0.5)

    btn1 = Button(root,text="Register",bg='black', fg='white',
command=studentRegister)
    btn1.place(relx=0.28,rely=0.3, relwidth=0.2,relheight=0.1)

    btn2 = Button(root,text="Login",bg='black', fg='white', command=Login)
    btn2.place(relx=0.53,rely=0.3, relwidth=0.2,relheight=0.1)

    btn3 = Button(root,text="Quit",bg='#455A64', fg='white', command=root.quit)
    btn3.place(relx=0.53,rely=0.9, relwidth=0.18,relheight=0.08)

# Take n greater than 0.25 and less than 5
same=True
n=0.3

# Adding a background image
background_image =Image.open("library.jpg")
[imageSizeWidth, imageSizeHeight] = background_image.size

newImageSizeWidth = int(imageSizeWidth*n)
if same:
    newImageSizeHeight = int(imageSizeHeight*n)
else:
    newImageSizeHeight = int(imageSizeHeight/n)

background_image = background_image.resize((newImageSizeWidth,newImageSizeHeight),Image.ANTIALIAS)
img = ImageTk.PhotoImage(background_image)

Canvas1 = Canvas(root)

Canvas1.create_image(300,340,image = img)
Canvas1.config(bg="white",width = newImageSizeWidth, height = newImageSizeHeight)
Canvas1.pack(expand=True,fill=BOTH)

headingFrame1 = Frame(root,bg="#333945",bd=5)
headingFrame1.place(relx=0.2,rely=0.1,relwidth=0.6,relheight=0.16)

headingFrame2 = Frame(headingFrame1,bg="#EAF0F1")
headingFrame2.place(relx=0.01,rely=0.05,relwidth=0.98,relheight=0.9)

headingLabel = Label(headingFrame2, text="Welcome to CMRCET Library", fg='black')
headingLabel.place(relx=0.25,rely=0.1, relwidth=0.5, relheight=0.5)

btn1 = Button(root,text="Employee",bg='black', fg='white', command=Employee)
btn1.place(relx=0.25,rely=0.3, relwidth=0.2,relheight=0.1)

btn2 = Button(root,text="Student",bg='black', fg='white', command=Student)
btn2.place(relx=0.55,rely=0.3, relwidth=0.2,relheight=0.1)

root.mainloop()
```

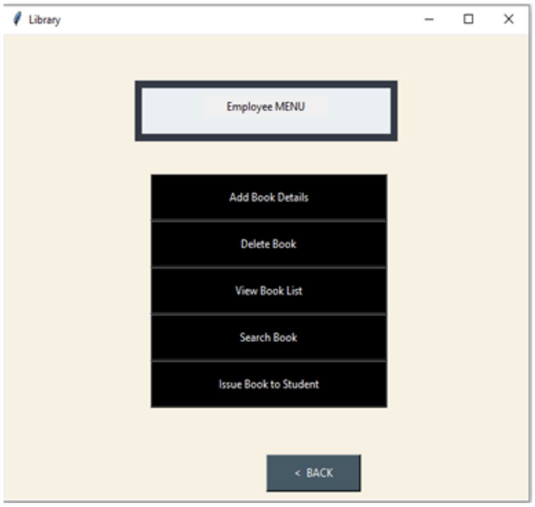
5.1 EXECUTION SCREENS

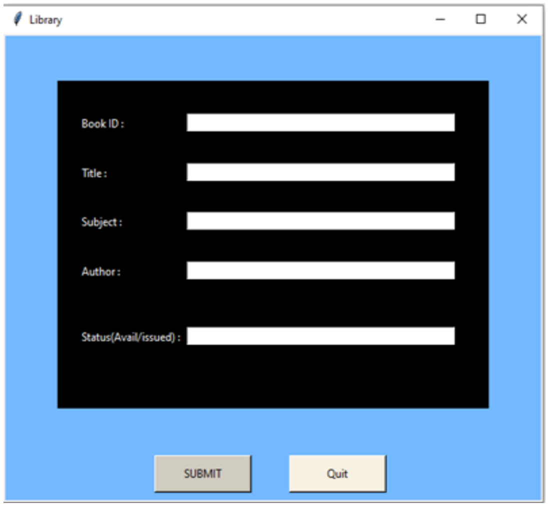


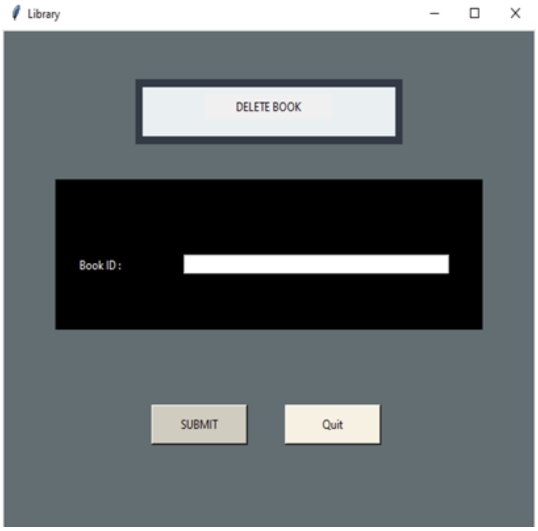


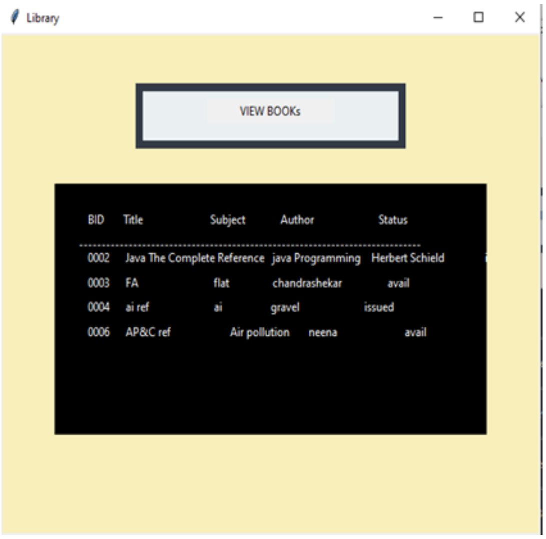




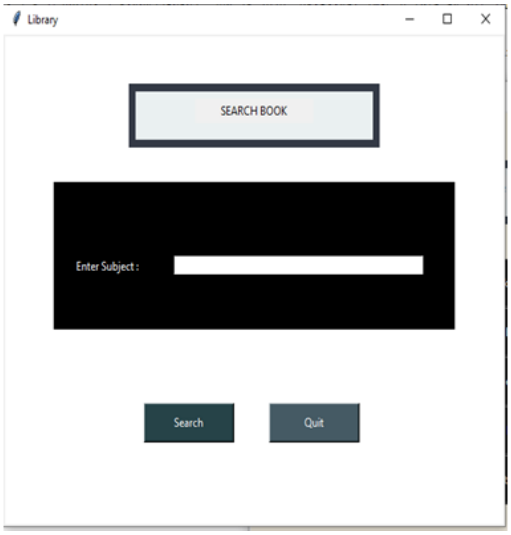


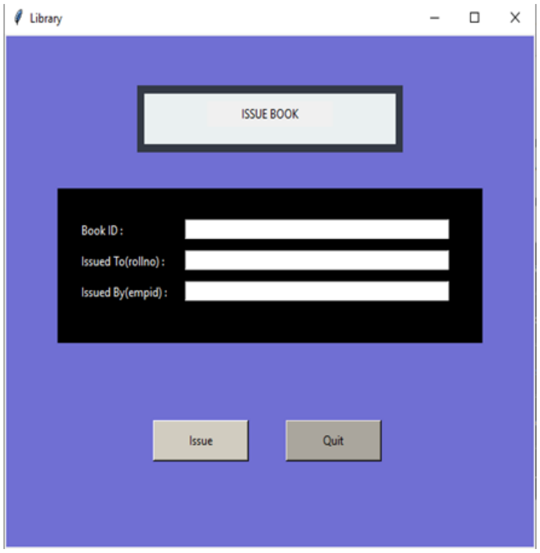


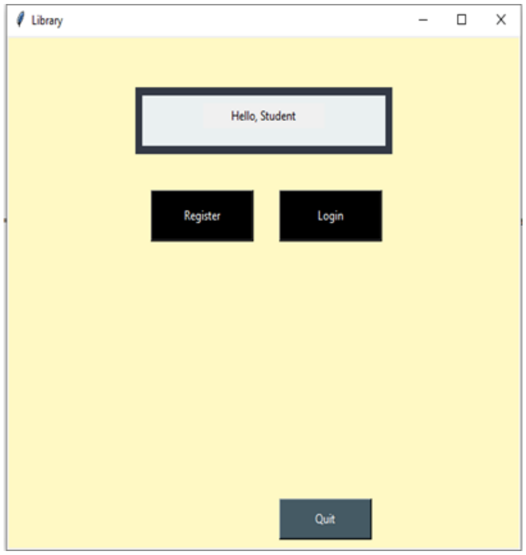


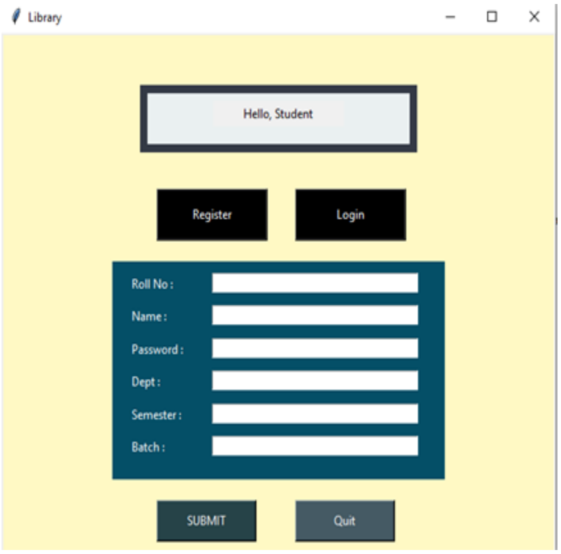


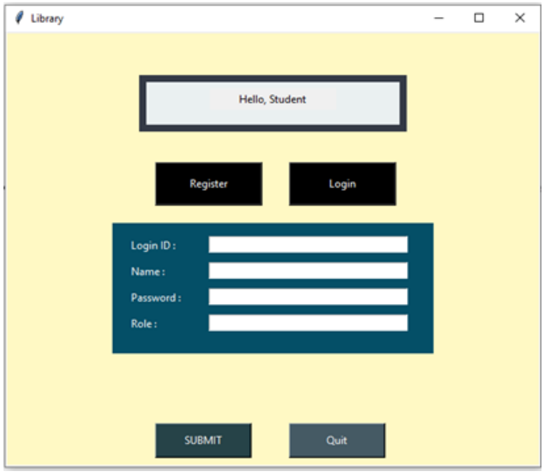
```
BID   Title                       Subject            Author              Status
-----------------------------------------------------------------------
0002  Java The Complete Reference java Programming   Herbert Schield         i
0003  FA                          flat             chandrashekar          avail
0004  ai ref                      ai              gravel               issued
0006  AP&C ref                        Air pollution   neena               avail
```

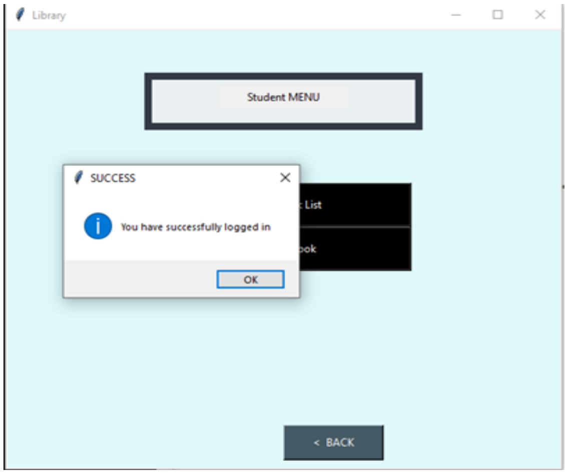




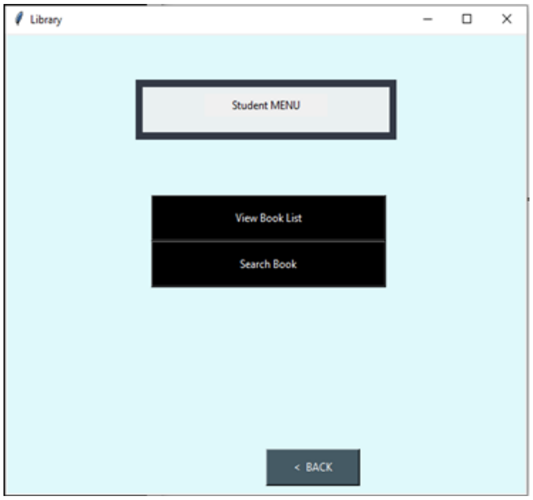


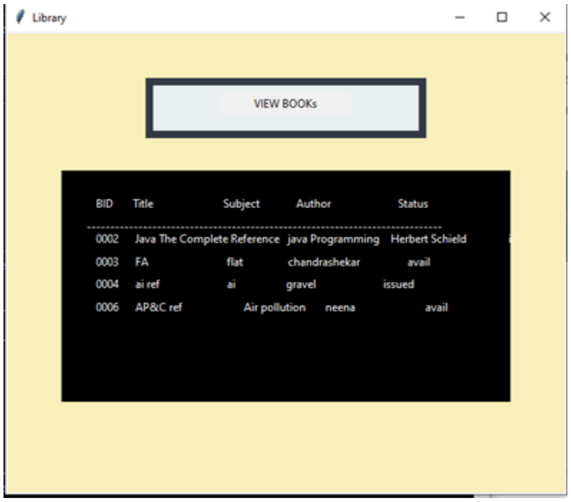




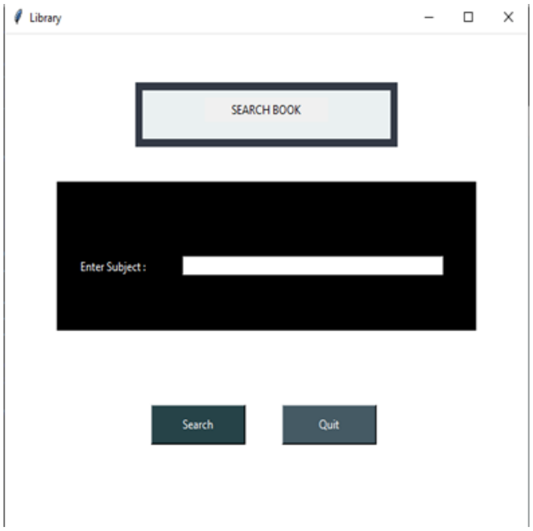


You have successfully logged in





```
BID   Title                      Subject          Author              Status
------------------------------------------------------------------------------
0002  Java The Complete Reference  java Programming  Herbert Schield        i
0003  FA                      flat           chandrashekar         avail
0004  ai ref                  ai            gravel              issued
0006  AP&C ref                   Air pollution    neena               avail
```



6. CONCLUSION

Computerized Library Management has been created keeping in mind the needs of Small and Medium scale libraries. It's efficient software that includes all the basic functionalities like making data entries for new books, registering a new user, editing and deleting records that are required for smooth functioning of a library. Additionally the user login and book history are also stored and can be accessed by the administrator. It also facilitates the librarian to create new user groups and Apart from this the users are also given the rights to not only keep track of the books ,can search for the books that interest them.

7. REFERENCES

1. Google for Problem solving
2. https://www.tutorialspont.com/um/uml modeling_types.htm
3. https://www.w3schools.com/html/
4. https://www.w3schools.com/sql/
5. https://www.wschools.com/php/default.asp
6. https://www.tutorilspoint.com/mysql/
7. https://www.educative.io/courses/grokking-the-object-oriented-design-interview/RMlM3NgjAyR