**BOOK BANK SYSTEM**

**1.Problem statement**

Now a day’s students should use library to borrow books. So, the book bank system is always maintaining the books, student records. Now in this Book Bank System have the details about the student’s login, book details they had now. The login details contain username and password of the students. These details are maintained in the database. The book details have the book name, author, quantity. The primary goal of the book bank system is to overcome the drawbacks of offline book system such as time and can’t check the book availability without visiting book bank. It is easy to maintain Book Bank System because most of the processes are automated.

**2. SYTEM REQUIREMENT SPECIFICATION:**

**2.1 Requirements:**

** User id, name (which is used to identify the admin and user)**

** Book name, Author (used to identify item)**

** Quantity, Date Difference (used to count and calculate the fine amount)**

**2.2 Project Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Module** | **Requirement** | **Plan** | **Risk** |
| 1 | Login/ Register | Username, Password | 1 | - |
| 2 | Add Book | Book name,  Author,  Quantity | 1 | - |
| 3 | Select Book | Book name,  Author,  Quantity | 1 | - |
| 4 | Return Book | Book name,  Author,  Quantity | 1 | - |
| 5 | Exit | - | 1 | - |

**2.3 Use Case Modelling:**

**2.3.1 Use Cases:**

A **use case** is a collection of related success and failure scenarios that describe actors using a system to support a goal.

The use cases involved in this project are,

* Login/Register
* Add Book
* Select Book
* Return Book
* Exit

**2.3.2 Actor:**

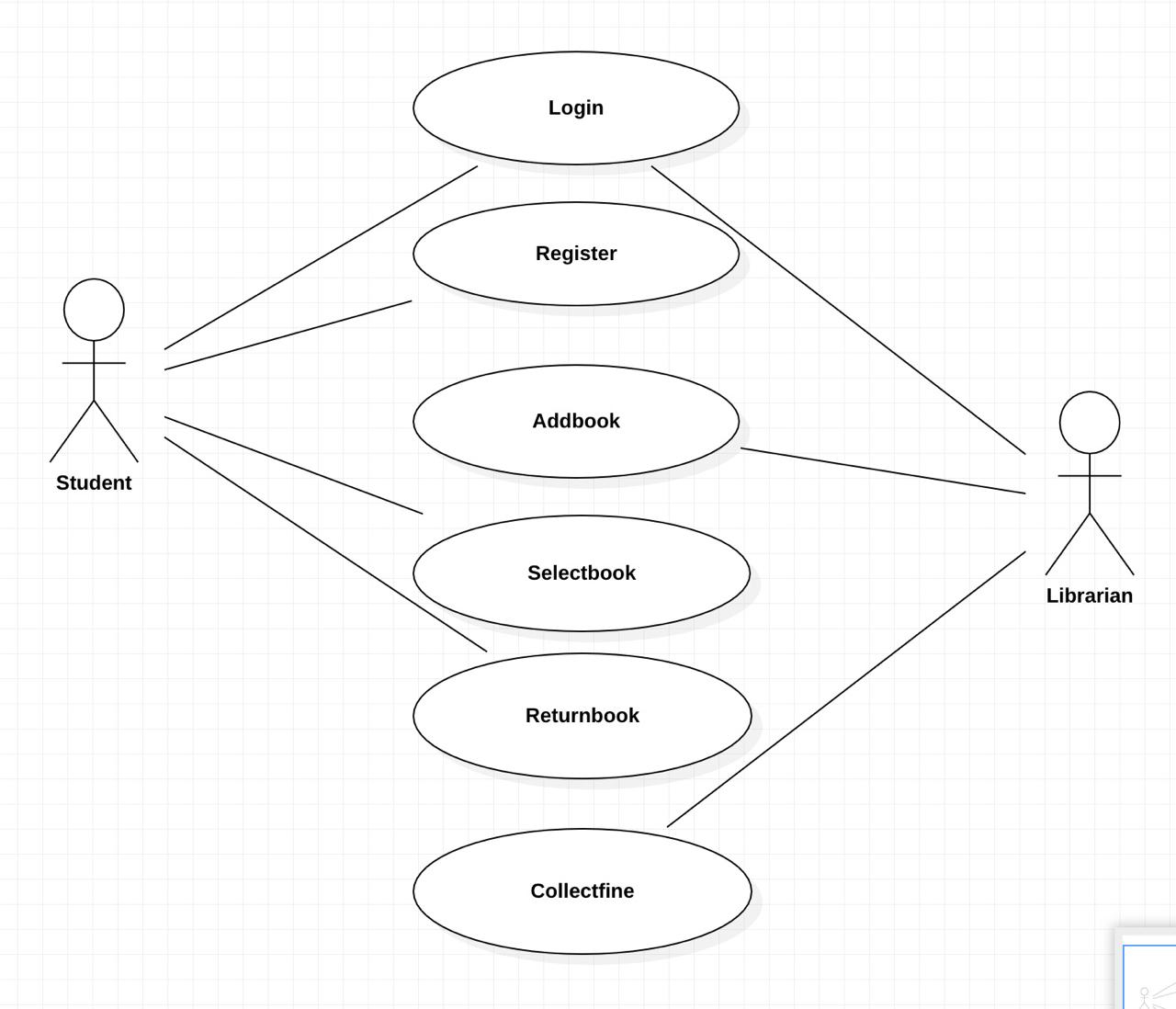
An **actor** is anything with behaviour, such as a person, or system or an organization, those who all involved in the Use case either directly or indirectly.

In this project the actors are,

* Librarian
* Student

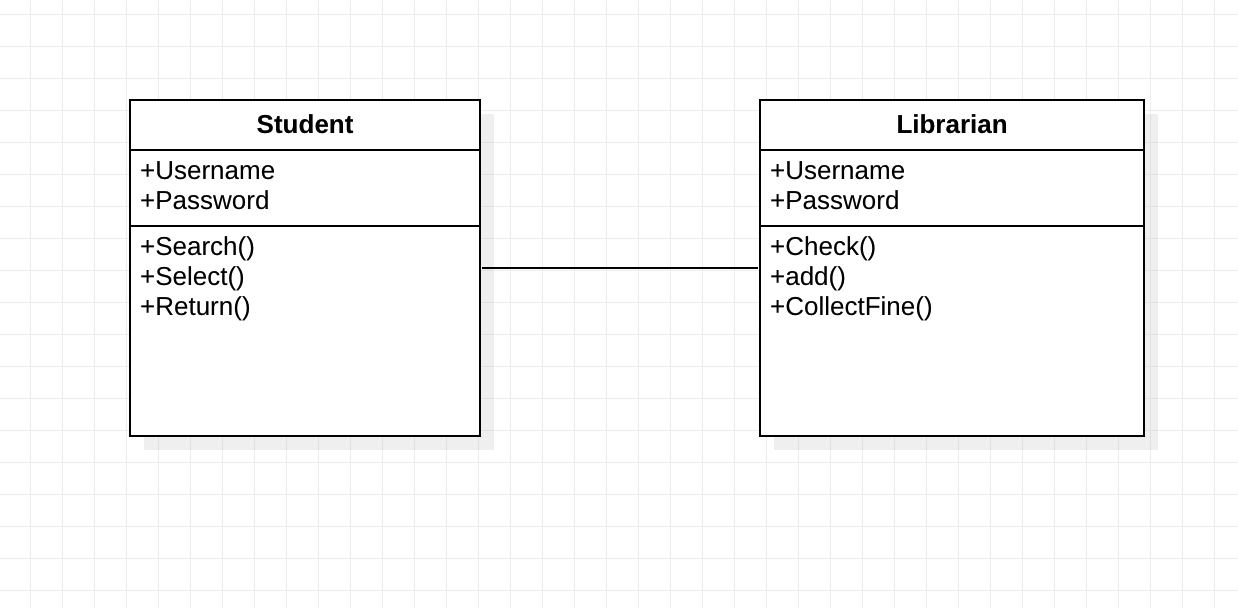
**2.3.3 Use Case Diagram:**

Use Case diagram describes the functionalities of a system from the user’s point of view.

****

**3. UML CLASS DIAGRAM:**

In software engineering, a **class diagram** in the Unified Modelling Language(UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or)methods and the relationships between the classes. A UML Class Box is used to illustrate software components often shows three compartments

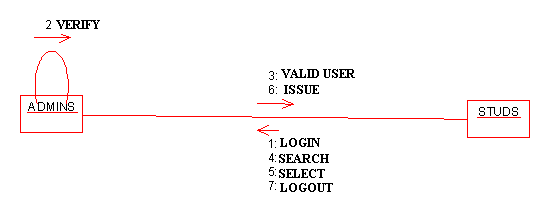


**4. INTERACTION DIAGRAMS:**

The term ***interaction diagram****,* is a generalization of two or more specialized UML diagram types; both can be used to express similar message interactions:

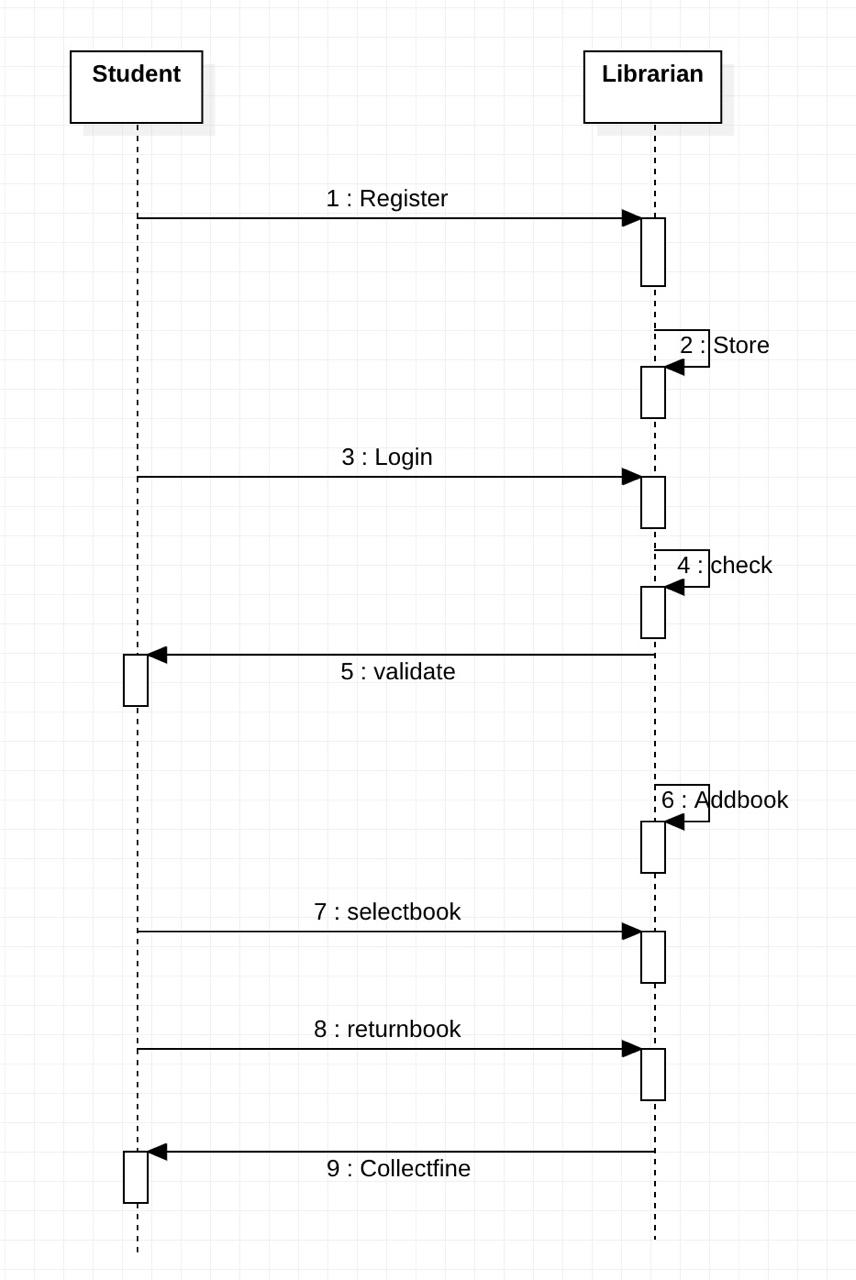
* Collaboration diagrams
* Sequence diagrams

**4.1 Collaboration Diagram:**



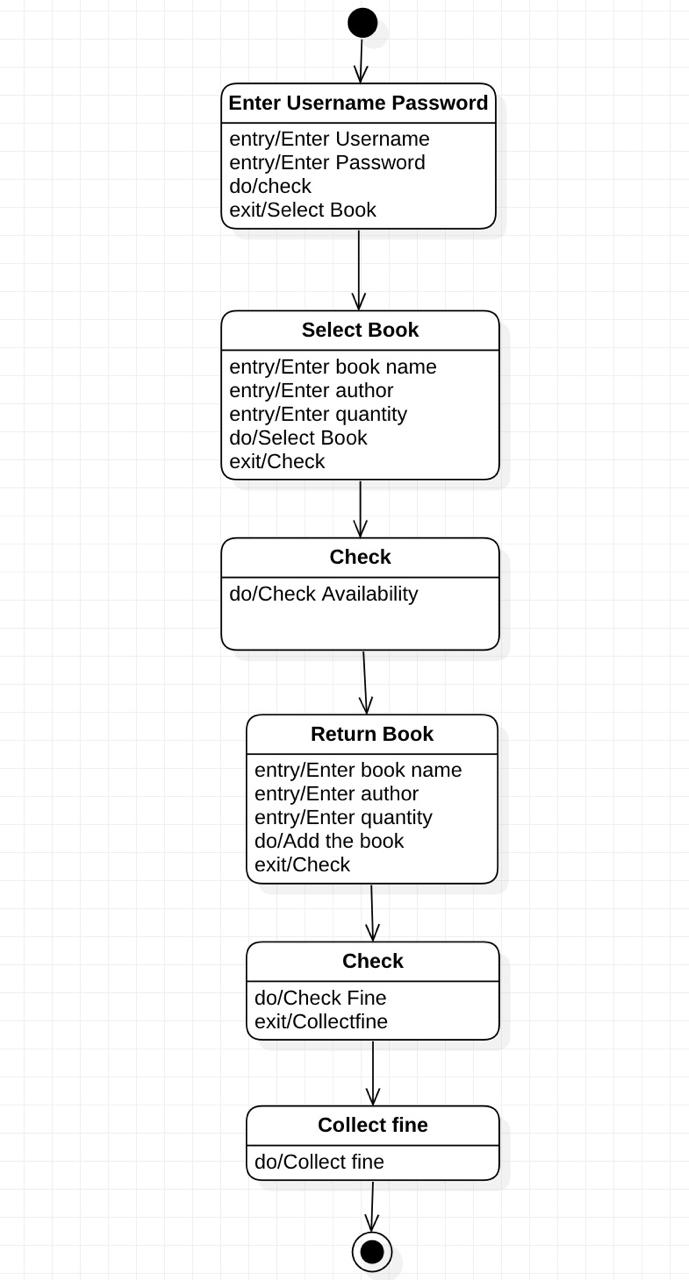
**4.2 Sequence Diagram:**

The UML includes **Sequence diagrams** as a notationthat can illustrate actor interactions and the operations initiated by them. A system sequence diagram is a picture that shows, for a particular scenario of use case, the events that external actors generate their order, and inter-system events. All systems are treated as a black box; the emphasis of the diagram is events that cross the system boundary from actors to systems

****

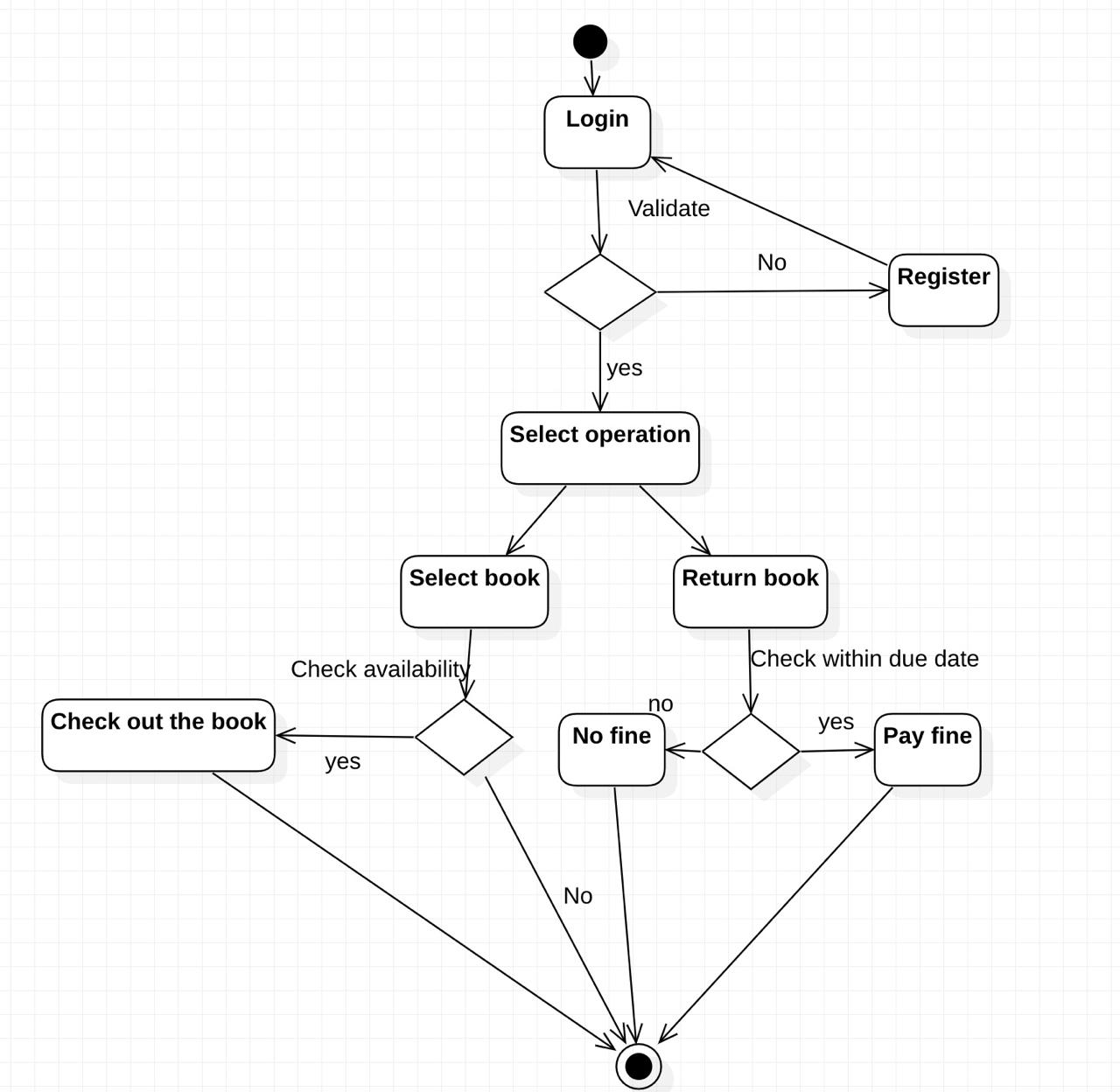
**5. STATE CHART DIAGRAM:**

A **state diagram** is a type of [diagram](http://en.wikipedia.org/wiki/Diagram) used in [computer science](http://en.wikipedia.org/wiki/Computer_science) and related fields to describe the behaviour of systems. State diagrams require that the system described is composed of a finite number of [states](http://en.wikipedia.org/wiki/State_(computer_science)); sometimes, this is indeed the case, while at other times this is a reasonable [abstraction](http://en.wikipedia.org/wiki/Abstraction_(computer_science)). There are many forms of state diagrams, which differ slightly and have different [semantics](http://en.wikipedia.org/wiki/Semantics#Computer_science).

****

**6.ACTIVITY DIAGRAM:**

In the Unified Modelling language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

****

**7. COMPONENT DIAGRAM:**

Component diagram is used to represent a software module with a

well defined interface



**8. LOGICAL ARCHITECHTURE REFINEMENT:**

A typical object-oriented information system is designed in terms of several architectural layers or subsystems as,

* User Interface – graphical interface; windows.
* Application logic and Domain objects – software objects representing domain concepts that full fill application requirements.
* Technical services – general purpose objects and subsystems that provide supporting technical services, such as interfacing with a database or error logging.

**Layer-1: User Interface layer:**

-Explore how to connect to other layers, and introduction to usability engineering issues.

-Minor focus.

In this project, it is implemented using Python GUI(Tkinter).

**Layer-2:** **Application logic and Domain Objects:**

-Explore how to design objects

-Primary focus of case study

**Layer-3: Technical Service layer:**

- Explore how to design objects

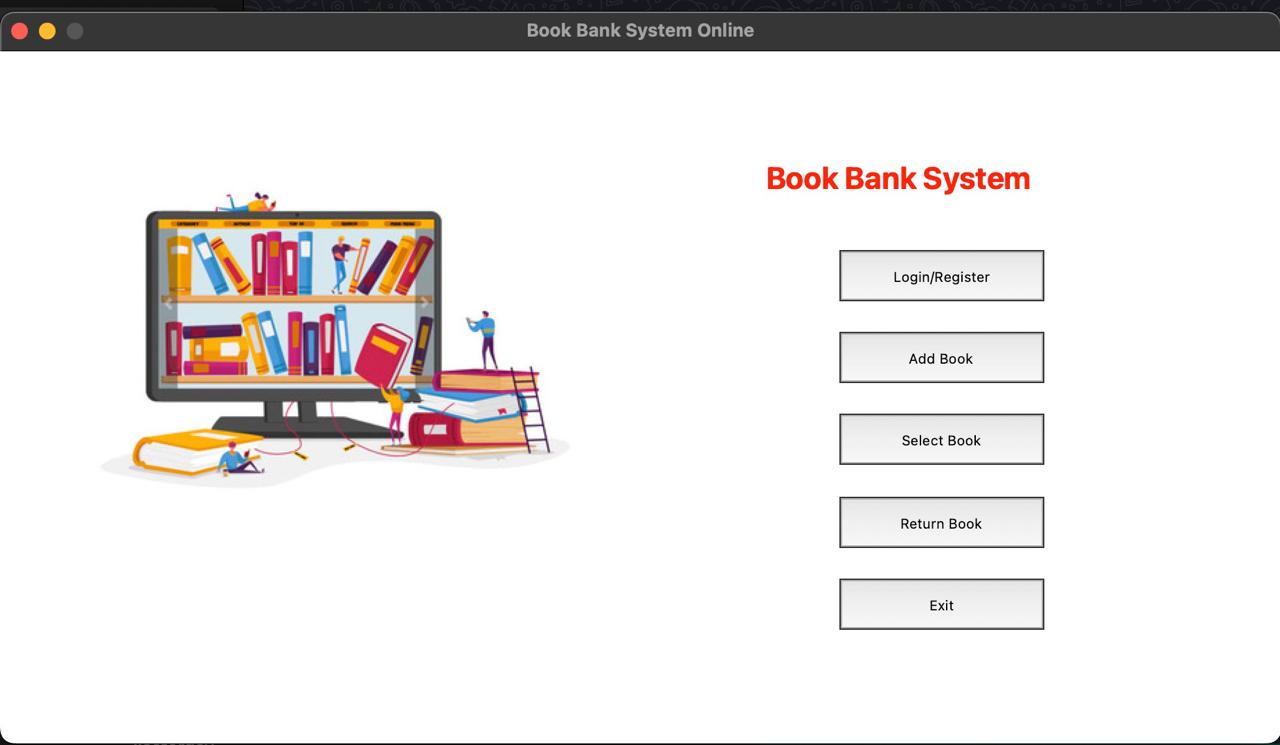
- Secondary focus

**8.1 Architectural Layer Diagram:**



**Front page**

**Home Page**

****

**def Entry\_page(self):**

**global root0**

**root0 = Tk()**

**root0.title("Book Bank System Online")**

**root0.geometry("925x500+200+100")**

**root0.resizable(False, False)**

**#root0.minsize(925, 500)**

**#root0.maxsize(1366, 768)**

**root0.configure(bg='#fff')**

**# Image**

**try:**

**img0 = PhotoImage(file="Front page.png")**

**bg\_img0 = Label(root0, image=img0, bg="white")**

**bg\_img0.place(x=65, y=85)**

**except Exception as e:**

**print(e)**

**# Frame for Buttons**

**frame0 = Frame(root0, width=370, height=370, bg='#fff')**

**frame0.place(x=480, y=70)**

**# Title Label**

**heading0 = Label(frame0, text="Book Bank System", fg='Red', bg='White', font=('Popins', 22, 'bold'))**

**heading0.place(x=70, y=5)**

**# Buttons**

**btn1 = Button(frame0, text="Login/Register", bg='orange', fg='black', border=0,font=('Popins', 10),command=self.signin\_1)**

**btn1.place(relx=0.34, rely=0.2, relwidth=0.40, relheight=0.10)**

**btn3 = Button(frame0, text="Add Book", bg='orange', fg='#000', border=0,font=('Popins', 10),command=self.log\_check3)**

**btn3.place(relx=0.34, rely=0.36, relwidth=0.40, relheight=0.10)**

**btn2 = Button(frame0, text="Select Book", bg='orange', fg='#000', border=0,font=('Popins', 10),command=self.log\_check1)**

**btn2.place(relx=0.34, rely=0.52, relwidth=0.40, relheight=0.10)**

**btn3 = Button(frame0, text="Return Book", bg='orange', fg='#000', border=0,font=('Popins', 10),command=self.log\_check2)**

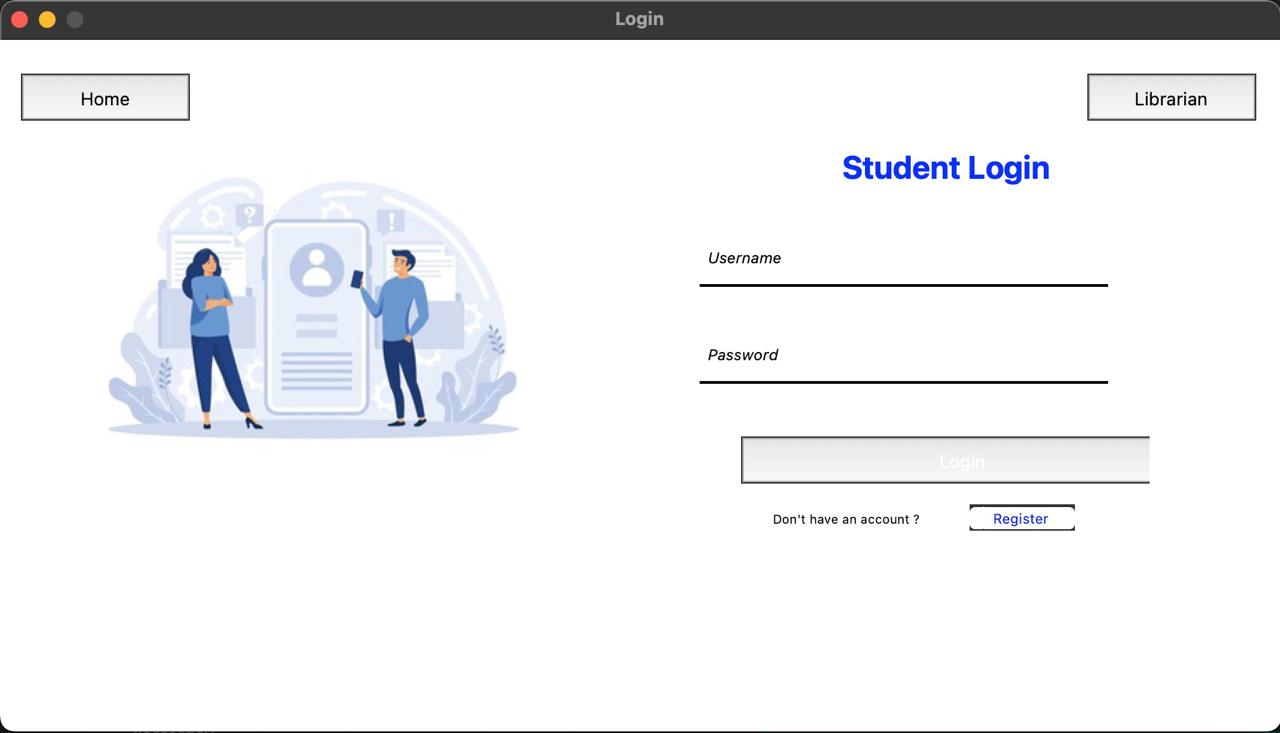
**btn3.place(relx=0.34, rely=0.68, relwidth=0.40, relheight=0.10)**

**btn4 = Button(frame0, text="Exit", bg='orange', fg='#000', command=root0.destroy,font=('Popins', 10), border=0)**

**btn4.place(relx=0.34, rely=0.84, relwidth=0.40, relheight=0.10)**

**root0.mainloop()**

**Student Login**

****

**def login(self):**

**global root1**

**root1 = Tk()**

**root1.title("Login")**

**root1.geometry("925x500+200+100")**

**root1.resizable(False, False)**

**root1.configure(bg='#fff')**

**def check():**

**username1=username.get()**

**passingword=Password.get()**

**if username1=='' or passingword=='' or username1=='Username' or passingword=='Password':**

**messagebox.showerror("Invalid","All feilds are required")**

**else:**

**try:**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**q="use userinfo"**

**cursor.execute(q)**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Sorry","Database connection failed")**

**return**

**query='select \*from userdata where username=%s and createpassword=%s'**

**cursor.execute(query,(username.get(),Password.get()))**

**row=cursor.fetchone()**

**if row==None:**

**messagebox.showerror('Error','Invalid username or password')**

**else:**

**global ma,ka**

**ma=True**

**ka=True**

**cursor.close()**

**con.close()**

**messagebox.showinfo('Login','Welcome to Book Bank System')**

**# Background Image**

**img = PhotoImage(file="Login page.png")**

**bg\_img = Label(root1, image=img, bg="white")**

**bg\_img.place(x=50, y=75)**

**# Frame for Login**

**frame = Frame(root1, width=350, height=350, bg='#fff')**

**frame.place(x=480, y=70)**

**# Sign in text**

**heading = Label(frame, text="Student Login", fg='blue', bg='White', font=('Popins', 23, 'bold'))**

**heading.place(x=125, y=5)**

**# Entry for Username**

**# on\_click function**

**def on\_click(key):**

**username.delete(0, END)**

**# on\_leave function**

**def on\_leave(key):**

**name = username.get()**

**if name == '':**

**username.insert(0, 'Username')**

**username = Entry(frame, width=25, fg='black', bg='white', highlightthickness=0, border=0, font=('Popins', 11, 'italic'))**

**username.place(x=30, y=80)**

**username.insert(0, 'Username')**

**username.bind('<FocusIn>', on\_click)**

**username.bind('<FocusOut>', on\_leave)**

**Frame(frame, width=295, height=2, bg='black').place(x=25, y=107)**

**# Entry for Password**

**# on\_click function**

**def on\_click(key):**

**Password.delete(0, END)**

**# on\_leave function**

**def on\_leave(key):**

**name = Password.get()**

**if name == '':**

**Password.insert(0, 'Password')**

**Password = Entry(frame, width=25, fg='black', bg='white', highlightthickness=0, border=0, font=('Popins', 11, 'italic'))**

**Password.place(x=30, y=150)**

**Password.insert(0, 'Password')**

**Frame(frame, width=295, height=2, bg='black').place(x=25, y=177)**

**Password.bind('<FocusIn>', on\_click)**

**Password.bind('<FocusOut>', on\_leave)**

**# Sign in Button**

**Button(frame, text="Login",width=32, pady=6, fg='#fff',bg='blue', border=0,command=check).place(x=55, y=217)**

**# new user? label**

**new\_user = Label(frame, text="Don't have an account ?", fg='black', bg='white', font=('Popins', 9))**

**new\_user.place(x=75, y=267)**

**# Register button**

**register = Button(frame, width=6, text='Register', bg='white', fg='blue', font=('Popins', 10), border=0, highlightthickness=0, cursor="hand2",command=self.register1)**

**register.place(x=220, y=266)**

**# Home Button**

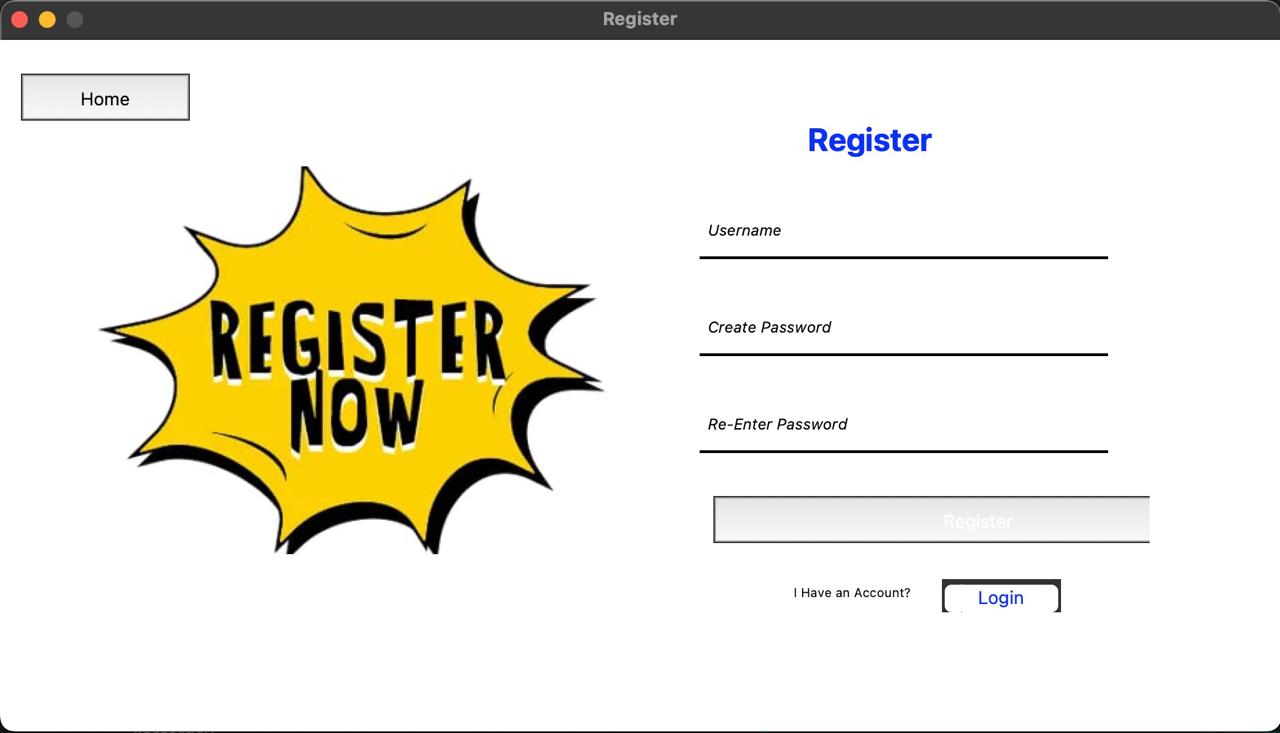
**Button(root1, text="Home",width=10, pady=6, fg='black',bg='orange', border=0,command=self.Entry\_page1).place(x=15, y=25)**

**# Librarian login**

**Button(root1, text="Librarian",width=10, pady=6, fg='black',bg='orange', border=0,command=self.lib\_login).place(x=835, y=25)**

**root1.mainloop()**

**Student Register:**

****

**def register(self):**

**global screen1**

**screen1 = Tk()**

**screen1.title("Register")**

**screen1.geometry("925x500+200+100")**

**screen1.resizable(False, False)**

**screen1.configure(bg='#fff')**

**# Background Image**

**img1 = Image.open("Register page.png")**

**img1 = ImageTk.PhotoImage(img1)**

**bg\_img1 = Label(screen1, image=img1, bg="white")**

**bg\_img1.place(x=50, y=90)**

**# Frame for Login**

**frame1 = Frame(screen1, width=350, height=390, bg='white')**

**frame1.place(x=480, y=50)**

**# Sign in text**

**heading = Label(frame1, text="Register", fg='blue', bg='White', font=('Popins', 23, 'bold'))**

**heading.place(x=100, y=5)**

**def get\_data():**

**username1=username.get()**

**CPassword1=CPassword.get()**

**RPassword1=RPassword.get()**

**if username1=='' or CPassword1=='' or RPassword1=='' or username1=='Username' or CPassword1=='Create Password' or RPassword1=='Re-Enter Password':**

**messagebox.showerror("Invalid","All feilds are required")**

**elif CPassword1!=RPassword1:**

**messagebox.showerror("Invalid","Password must be same")**

**else:**

**try:**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Sorry","Database connection failed")**

**return**

**try:**

**q="create database userinfo"**

**cursor.execute(q)**

**q="use userinfo"**

**cursor.execute(q)**

**except Exception as e:**

**print(e,2)**

**try:**

**cursor.execute("use userinfo")**

**q="create table userdata(username varchar(20),createpassword varchar(20),reenteredpassword varchar(20),primary key(username))"**

**cursor.execute(q)**

**except Exception as e:**

**print(e,'table')**

**try:**

**q="insert into userdata(username,createpassword,reenteredpassword) values(%s,%s,%s)"**

**cursor.execute(q,(username1,CPassword1,RPassword1))**

**k=True**

**except Exception as e:**

**print(e,3)**

**messagebox.showerror("Error","username is already taken.")**

**con.commit()**

**cursor.close()**

**con.close()**

**try:**

**if k==True:**

**messagebox.showinfo("Registration","Registration Sucessful")**

**except:**

**messagebox.showerror("Error","Try again")**

**# Entry for Username**

**# on\_click function**

**def on\_click(key):**

**username.delete(0, END)**

**# on\_leave function**

**def on\_leave(key):**

**name = username.get()**

**if name == '':**

**username.insert(0, 'Username')**

**username = Entry(frame1, width=25, fg='black', bg='white', highlightthickness=0, border=0,**

**font=('Popins', 11, 'italic'))**

**username.place(x=30, y=80)**

**username.insert(0,'Username')**

**username.bind('<FocusIn>', on\_click)**

**username.bind('<FocusOut>', on\_leave)**

**Frame(frame1, width=295, height=2, bg='black').place(x=25, y=107)**

**# Entry for Create Password**

**# on\_click function**

**def on\_click(key):**

**CPassword.delete(0, END)**

**# on\_leave function**

**def on\_leave(key):**

**name = CPassword.get()**

**if name == '':**

**CPassword.insert(0, 'Create Password')**

**CPassword = Entry(frame1, width=25, fg='black', bg='white', highlightthickness=0, border=0,**

**font=('Popins', 11, 'italic'))**

**CPassword.place(x=30, y=150)**

**CPassword.insert(0, 'Create Password')**

**Frame(frame1, width=295, height=2, bg='black').place(x=25, y=177)**

**CPassword.bind('<FocusIn>', on\_click)**

**CPassword.bind('<FocusOut>', on\_leave)**

**# Entry for Re-enter Password**

**# on\_click function**

**def on\_click(key):**

**RPassword.delete(0, END)**

**# on\_leave function**

**def on\_leave(key):**

**name = RPassword.get()**

**if name == '':**

**RPassword.insert(0, 'Re-Enter Password')**

**RPassword = Entry(frame1, width=25, fg='black', bg='white', highlightthickness=0, border=0,**

**font=('Popins', 11, 'italic'))**

**RPassword.place(x=30, y=220)**

**RPassword.insert(0, 'Re-Enter Password')**

**Frame(frame1, width=295, height=2, bg='black').place(x=25, y=247)**

**RPassword.bind('<FocusIn>', on\_click)**

**RPassword.bind('<FocusOut>', on\_leave)**

**# Register Button**

**Button(frame1, text="Register", width=39, pady=6, fg='white',bg='blue', border=0,command=get\_data).place(x=35, y=280)**

**# Already Have an Account? label**

**label=Label(frame1,text="I Have an Account?",fg="black",bg="white",font=('Popins', 9))**

**label.place(x=90,y=340)**

**# Sign in button**

**sign\_in = Button(frame1, width=6, text='Login', bg='white', fg='blue',border=0, cursor='hand2',command=self.Entry\_page5)**

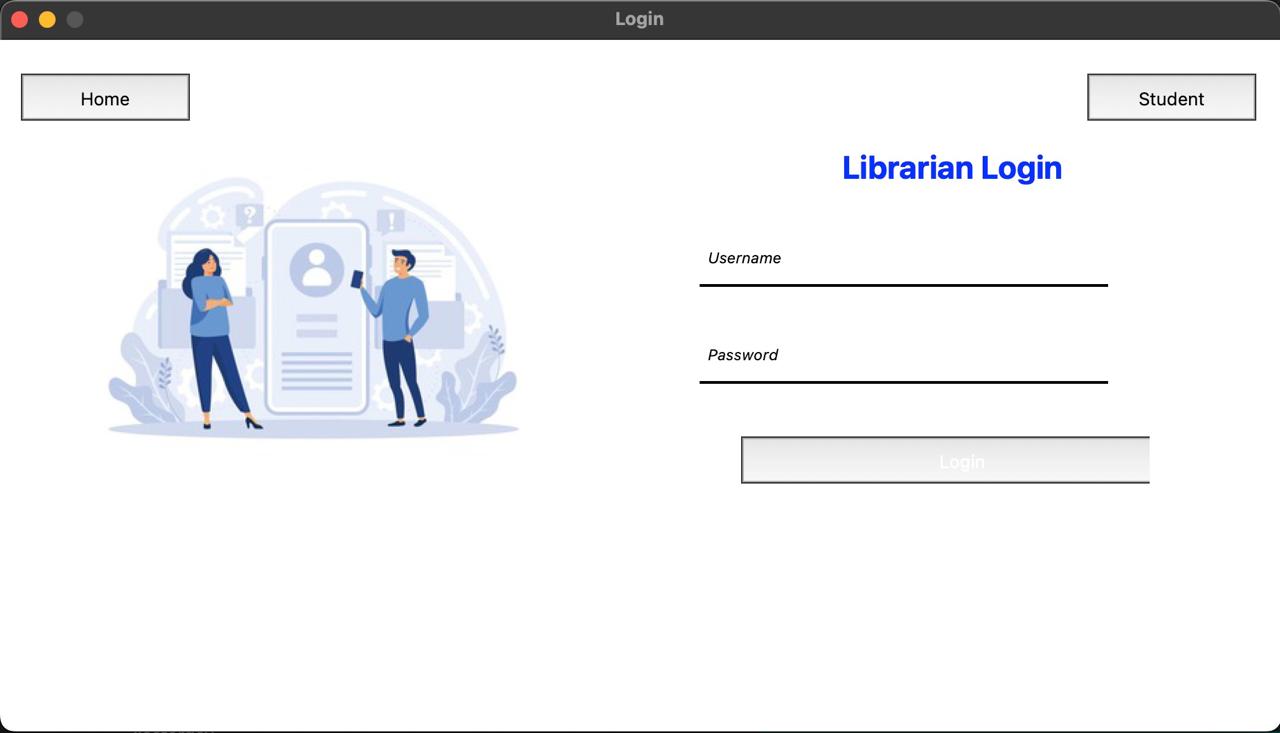
**sign\_in.place(x=200, y=340)**

**# Home Button**

**Button(screen1, text="Home",width=10, pady=6, fg='black',bg='orange', border=0,command=self.Entry\_page5).place(x=15, y=25)**

**screen1.mainloop()**

**Librarian Login:**

****

**def librarian\_login\_page(self):**

**global root3**

**root3 = Tk()**

**root3.title("Login")**

**root3.geometry("925x500+200+100")**

**root3.resizable(False, False)**

**root3.configure(bg='#fff')**

**def checking():**

**username1=username.get()**

**passingword=Password.get()**

**if username1=='' or passingword=='' or username1=='Username' or passingword=='Password':**

**messagebox.showerror("Invalid","All feilds are required")**

**elif username1=='Admin' and passingword=='root@1':**

**global la**

**la=True**

**messagebox.showinfo('Login','Welcome to Book Bank System')**

**else:**

**messagebox.showerror('Error','Invalid username or password')**

**img = PhotoImage(file="Login page.png")**

**bg\_img = Label(root3, image=img, bg="white")**

**bg\_img.place(x=50, y=75)**

**# Frame for Login**

**frame = Frame(root3, width=350, height=350, bg='#fff')**

**frame.place(x=480, y=70)**

**# Sign in text**

**heading = Label(frame, text="Librarian Login", fg='blue', bg='White', font=('Popins', 23, 'bold'))**

**heading.place(x=125, y=5)**

**# Entry for Username**

**# on\_click function**

**def on\_click(key):**

**username.delete(0, END)**

**# on\_leave function**

**def on\_leave(key):**

**name = username.get()**

**if name == '':**

**username.insert(0, 'Username')**

**username = Entry(frame, width=25, fg='black', bg='white', highlightthickness=0, border=0, font=('Popins', 11, 'italic'))**

**username.place(x=30, y=80)**

**username.insert(0, 'Username')**

**username.bind('<FocusIn>', on\_click)**

**username.bind('<FocusOut>', on\_leave)**

**Frame(frame, width=295, height=2, bg='black').place(x=25, y=107)**

**# Entry for Password**

**# on\_click function**

**def on\_click(key):**

**Password.delete(0, END)**

**# on\_leave function**

**def on\_leave(key):**

**name = Password.get()**

**if name == '':**

**Password.insert(0, 'Password')**

**Password = Entry(frame, width=25, fg='black', bg='white', highlightthickness=0, border=0, font=('Popins', 11, 'italic'))**

**Password.place(x=30, y=150)**

**Password.insert(0, 'Password')**

**Frame(frame, width=295, height=2, bg='black').place(x=25, y=177)**

**Password.bind('<FocusIn>', on\_click)**

**Password.bind('<FocusOut>', on\_leave)**

**# Sign in Button**

**Button(frame, text="Login",width=32, pady=6, fg='#fff',bg='blue', border=0,command=checking).place(x=55, y=217)**

**# Home Button**

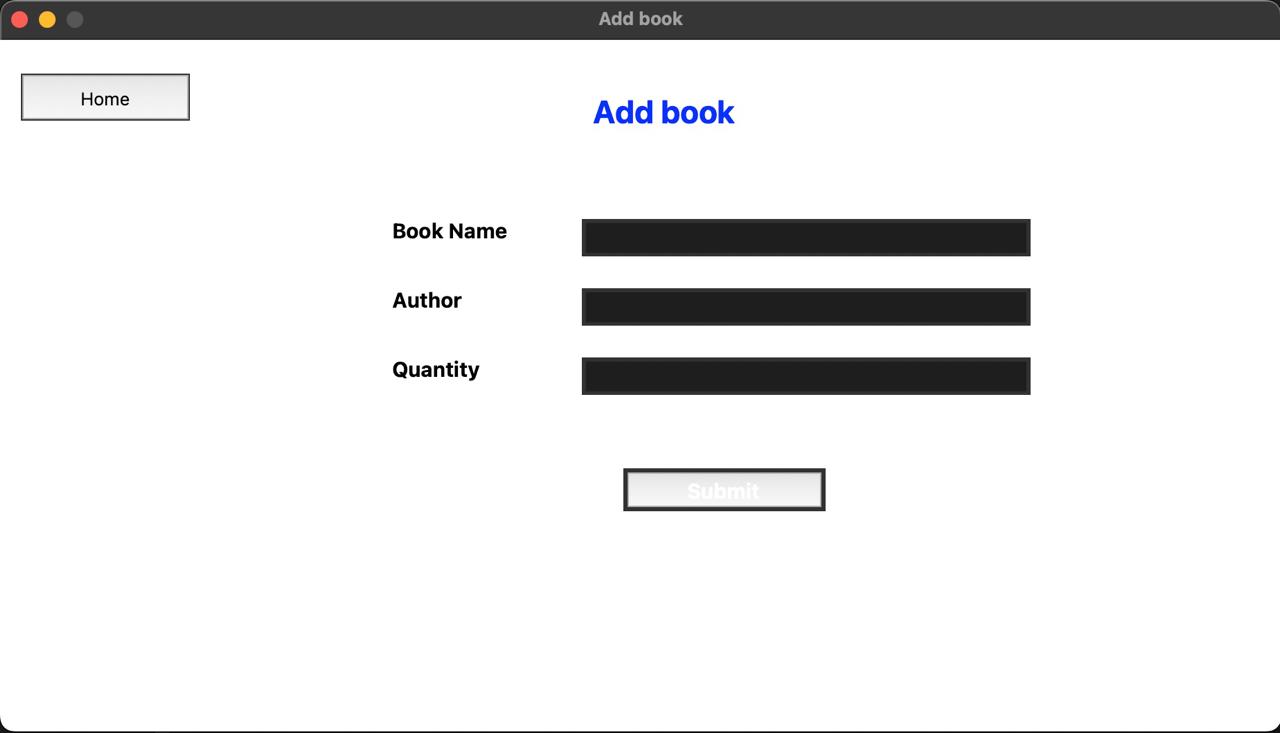
**Button(root3, text="Home",width=10, pady=6, fg='black',bg='orange', border=0,command=self.Entry\_page6).place(x=15, y=25)**

**# Student login**

**Button(root3, text="Student",width=10, pady=6, fg='black',bg='orange', border=0,command=self.stu\_login).place(x=835, y=25)**

**root3.mainloop()**

**Add Book**

****

**def add\_book(self):**

**global screen4**

**screen4 = Tk()**

**screen4.geometry("925x500+200+100")**

**screen4.resizable(False, False)**

**screen4.configure(bg='#fff')**

**screen4.title("Add book")**

**frame4=Frame(screen4,width=500,height=400,bg='#fff')**

**frame4.place(x=250, y=30)**

**heading = Label(frame4, text="Add book", fg='blue', bg='White', font=('Popins', 23, 'bold'))**

**heading.place(x=175, y=5)**

**Label(frame4, text = "Book Name ",font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 30,y = 95)**

**Label(frame4, text = "Author ", font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 30, y = 145)**

**Label(frame4,text = "Quantity ", font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 30, y = 195)**

**sbmitbtn = Button(frame4,width=10, text = "Submit",font=('popins',15,'bold'),bg = "blue", fg = "white",command=self.add\_bo).place(x = 200, y = 280)**

**global bu1,au1,qu1**

**bu1 = StringVar()**

**au1 = StringVar()**

**qu1 = StringVar()**

**bn = Entry(frame4,textvariable=bu1,width=39,font=('popins',12)).place(x = 170, y = 100)**

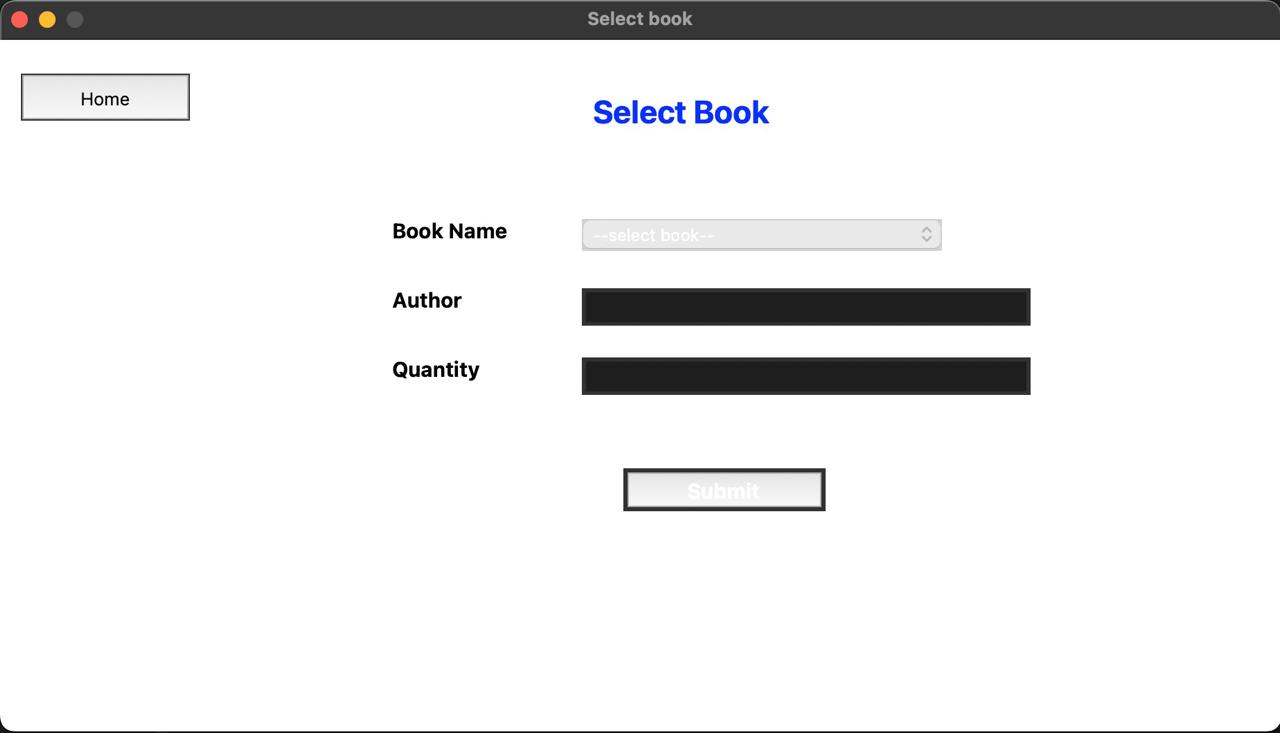
**au = Entry(frame4,textvariable=au1,width=39,font=('popins',12)).place(x = 170, y = 150)**

**qu = Entry(frame4,textvariable=qu1,width=39,font=('popins',12)).place(x = 170, y = 200)**

**Button(screen4, text="Home",width=10, pady=6, fg='black',bg='orange', border=0,command=self.Entry\_page7).place(x=15, y=25)**

**screen4.mainloop()**

**Select book:**

****

**def select\_book(self):**

**global screen2**

**screen2 = Tk()**

**screen2.geometry("925x500+200+100")**

**screen2.resizable(False, False)**

**screen2.configure(bg='#fff')**

**screen2.title("Select book")**

**frame2=Frame(screen2,width=500,height=400,bg='#fff')**

**frame2.place(x=250, y=30)**

**heading = Label(frame2, text="Select Book", fg='blue', bg='White', font=('Popins', 23, 'bold'))**

**heading.place(x=175, y=5)**

**Label(frame2, text = "Book Name ",font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 30,y = 95)**

**Label(frame2, text = "Author ", font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 30, y = 145)**

**Label(frame2,text = "Quantity ", font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 30, y = 195)**

**sbmitbtn = Button(frame2,width=10, text = "Submit",font=('popins',15,'bold'),bg = "blue", fg = "white",command=self.sel\_bo).place(x = 200, y = 280)**

**option=[]**

**try:**

**global con,cursor**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**q='use userinfo'**

**cursor.execute(q)**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Invalid","Database connection failed")**

**return**

**q='select bookname from book'**

**cursor.execute(q)**

**row=cursor.fetchall()**

**for k in row:**

**option.append(str(k[0]))**

**global var,author1,quantity1**

**var=StringVar()**

**var.set('--select book--')**

**author1=StringVar()**

**quantity1=StringVar()**

**b\_name = OptionMenu(frame2,var,\*option)**

**b\_name.configure(width=27,font=('popins',12),bg='#d3d3d3')**

**b\_name.place(x = 170, y = 100)**

**b\_name.config(width=27,font=('popins',12),bg='#d3d3d3')**

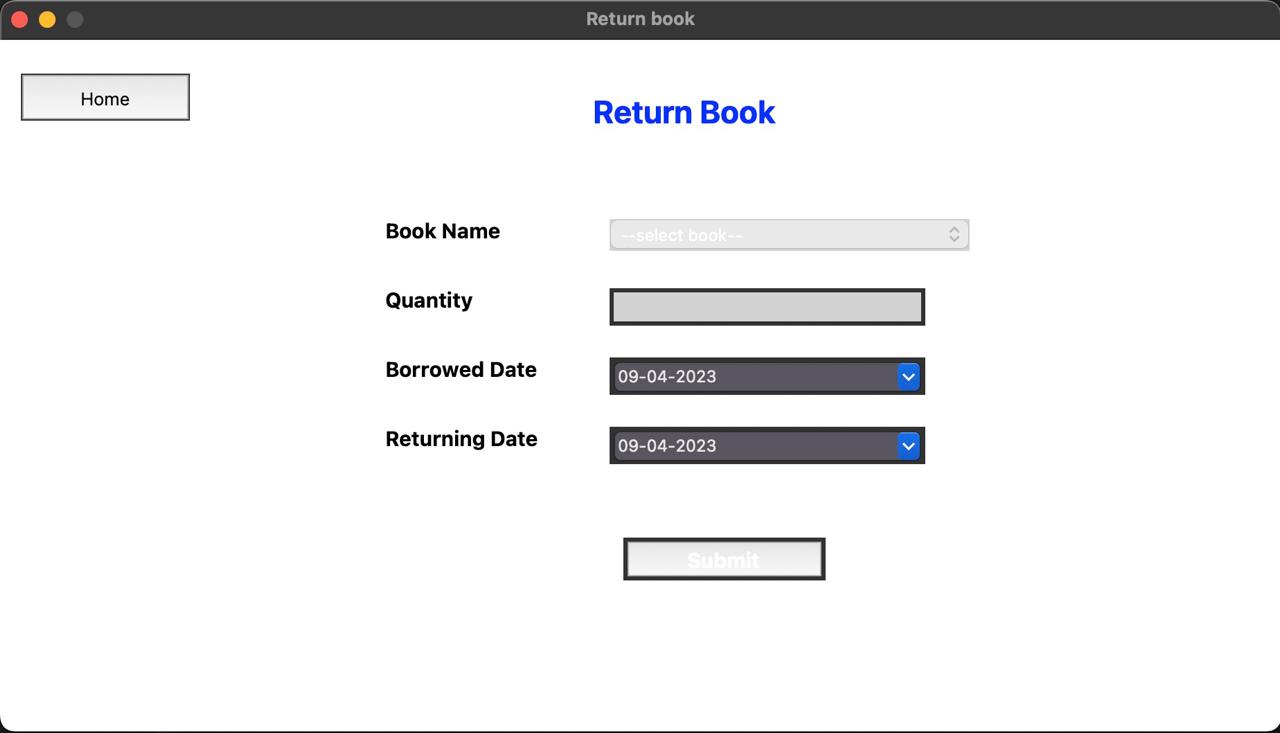
**author = Entry(frame2,textvariable=author1,width=39,font=('popins',12)).place(x = 170, y = 150)**

**quantity = Entry(frame2,textvariable=quantity1,width=39,font=('popins',12)).place(x = 170, y = 200)**

**Button(screen2, text="Home",width=10, pady=6, fg='black',bg='orange', border=0,command=self.Entry\_page2).place(x=15, y=25)**

**screen2.mainloop()**

**Return Book**

****

**def return\_book(self):**

**global screen3**

**screen3 = Tk()**

**screen3.geometry("925x500+200+100")**

**screen3.resizable(False, False)**

**screen3.configure(bg='#fff')**

**screen3.title("Return book")**

**frame3=Frame(screen3,width=500,height=400,bg='#fff')**

**frame3.place(x=250, y=30)**

**heading = Label(frame3, text="Return Book", fg='blue', bg='White', font=('Popins', 23, 'bold'))**

**heading.place(x=175, y=5)**

**Label(frame3, text = "Book Name ",font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 25,y = 95)**

**Label(frame3, text = "Quantity ", font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 25, y = 145)**

**Label(frame3,text ="Borrowed Date ", font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 25, y = 195)**

**Label(frame3,text = "Returning Date ", font=('popins',15,'bold'),bg='#fff',fg='black').place(x = 25, y = 245)**

**option1=[]**

**try:**

**global con,cursor**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**q='use userinfo'**

**cursor.execute(q)**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Invalid","Database connection failed")**

**return**

**q='select bookname from book'**

**cursor.execute(q)**

**row=cursor.fetchall()**

**for k in row:**

**option1.append(str(k[0]))**

**global bor\_date,bk\_quntity**

**global ret\_date**

**global bo\_name**

**bo\_name=StringVar()**

**bo\_name.set('--select book--')**

**bk\_quntity=StringVar()**

**bor\_date=StringVar()**

**ret\_date=StringVar()**

**b\_name = OptionMenu(frame3,bo\_name,\*option1)**

**b\_name.configure(width=27,font=('popins',12),bg='#d3d3d3')**

**b\_name.place(x = 190, y = 100)**

**b\_qua = Entry(frame3,width=27,font=('popins',12),bg='#d3d3d3',textvariable=bk\_quntity).place(x = 190, y = 150)**

**b\_date = DateEntry(frame3,selectmode='day',date\_pattern="dd-MM-yyyy",width=25,font=('popins',12),bg='#d3d3d3',textvariable=bor\_date).place(x = 190, y = 200)**

**r\_date = DateEntry(frame3,selectmode='day',date\_pattern="dd-MM-yyyy",width=25,font=('popins',12),bg='#d3d3d3',textvariable=ret\_date).place(x = 190, y = 250)**

**sbmitbtn2 = Button(frame3,width=10, text = "Submit",font=('popins',15,'bold'),bg = "blue", fg = "white",command=self.ret).place(x = 200, y = 330)**

**Button(screen3, text="Home",width=10, pady=6, fg='black',bg='orange', border=0,command=self.Entry\_page3).place(x=15, y=25)**

**screen3.mainloop()**

**8.2 Technical Service Layer:**

**Back-end:** MySQL

**Front-end:** Python GUI(Tkinter)

**8.2.1 Steps for creating database & recordset:**

1. Apache server 🡪 Phpmyadmin 🡪 Create new database 🡪 Enter name
2. Build a table by adding necessary fields.

Database Name: userinfo

Table Name : userdata ,book

**userdata:**

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | useraname | createpassword | reenterpassword |
| 1 | mahi | 123 | 123 |

**For Create and Insert records on userdata:**

**try:**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Sorry","Database connection failed")**

**return**

**try:**

**q="create database userinfo"**

**cursor.execute(q)**

**q="use userinfo"**

**cursor.execute(q)**

**except Exception as e:**

**print(e,2)**

**try:**

**cursor.execute("use userinfo")**

**q="create table userdata(username varchar(20),createpassword varchar(20),reenteredpassword varchar(20),primary key(username))"**

**cursor.execute(q)**

**except Exception as e:**

**print(e,'table')**

**try:**

**q="insert into userdata(username,createpassword,reenteredpassword) values(%s,%s,%s)"**

**cursor.execute(q,(username1,CPassword1,RPassword1))**

**k=True**

**except Exception as e:**

**print(e,3)**

**messagebox.showerror("Error","username is already taken.")**

**con.commit()**

**cursor.close()**

**con.close()**

**For verify login in userdata:**

**try:**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**q="use userinfo"**

**cursor.execute(q)**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Sorry","Database connection failed")**

**return**

**query='select \*from userdata where username=%s and createpassword=%s'**

**cursor.execute(query,(username.get(),Password.get()))**

**row=cursor.fetchone()**

**if row==None:**

**messagebox.showerror('Error','Invalid username or password')**

**book:**

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | bookname | author | quantity |
| 1 | C++ | Mahi | 20 |
| 2 | Data Structure | Murugan | 10 |

**Create and Insert records on book:**

**try:**

**global con,cursor**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**q='use userinfo'**

**cursor.execute(q)**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Invalid","Database connection failed")**

**try:**

**q='create table book(bookname varchar(20),author varchar(20),quantity I nt,primary key(bookname))'**

**cursor.execute(q)**

**except Exception as e:**

**print(e,'Table exists')**

**b1=bu1.get()**

**a1=au1.get()**

**q1=qu1.get()**

**q1=int(q1)**

**try:**

**global con,cursor**

**con=db.connect(host="localhost",user="root",password='')**

**cursor=con.cursor()**

**q='use userinfo'**

**cursor.execute(q)**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Invalid","Database connection failed")**

**return**

**try:**

**q='insert into book(bookname,author,quantity) values(%s,%s,%s)'**

**cursor.execute(q,(b1,a1,q1))**

**messagebox.showinfo("Addbook","{} book is sucessfully added".format(b1))**

**except Exception as e:**

**print(e,1)**

**messagebox.showerror("Error","Can't insert book")**

**con.commit()**

**cursor.close()**

**con.close()**

**8.2.2 Steps for establishing Connection between front-end and back-end:**

1. Go to Xampp 🡪 Turn on MySQL and Apache server 🡪Install mysql.connector 🡪 Import mysql.connector and create cursor .
2. To use the database with python to achieve the connection between frontend and backend.
3. Assign the data fields accordingly to the corresponding entries.

**Conclusion:**

In this project we can maintain the login and book details in an easy manner without taking more efforts. Thus, the Maintenance of the Book Bank System also very easy.