LIBRARY MANAGEMENT SYSTEM

BY :-
RAMYA P (220701217)
RAJAKUMARAN BHAVANISHRAJ (220701215)

Table Of Content

- SOFTWARE DETAILS
- ER - DIAGRAM
- ABSTRACT
- IMPLEMENTATION

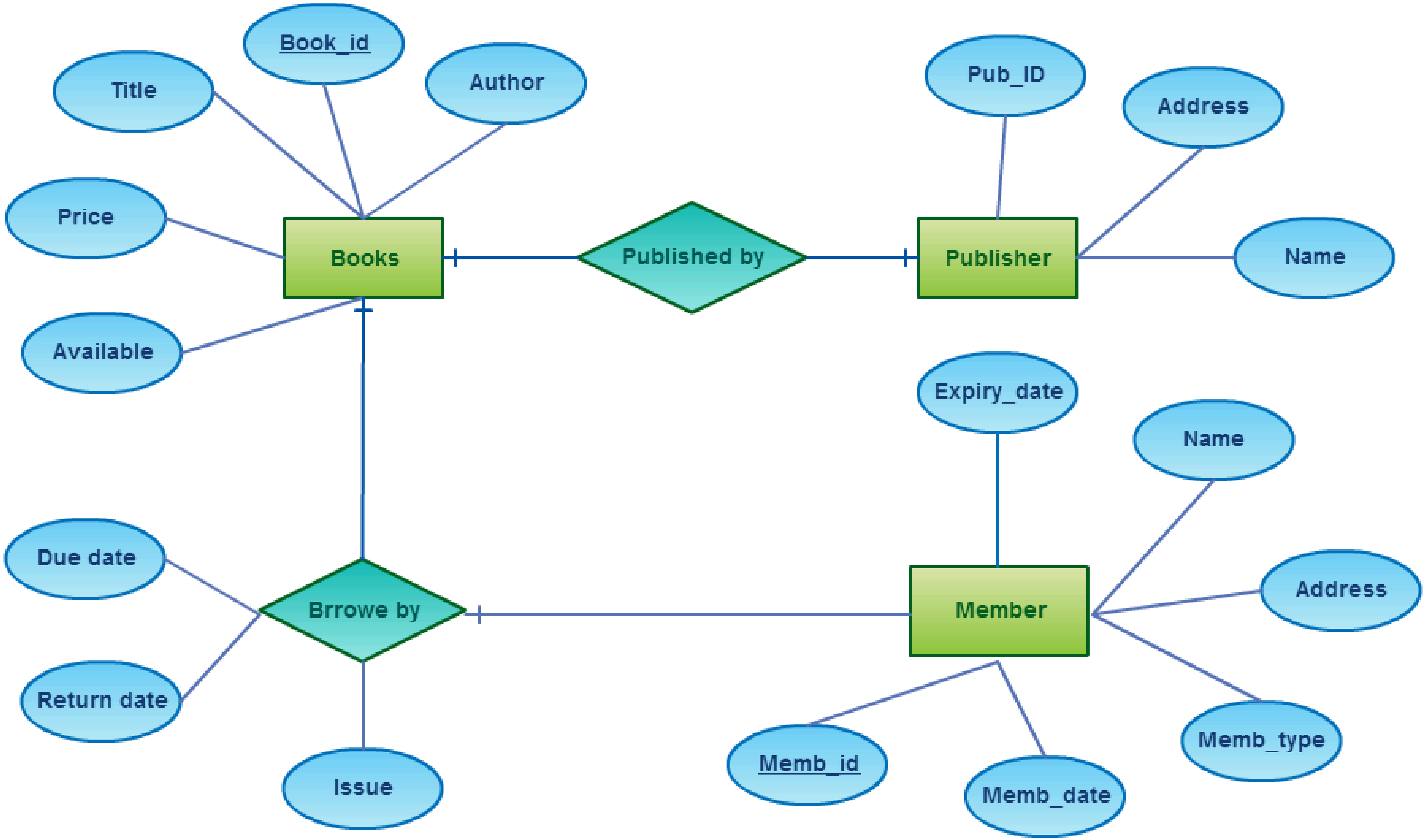
SOFTWARE DETAILS

SOFTWARE DETAILS

- Front End Tool Used in this project is “PYTHON”.
- Database Connectivity is done with the Front End Tool.
- Backend Tool Used Here in this project is “MySQL”.

ENTITY - RELATIONSHIP DIAGRAM

E-R Diagram of Library Management System



ABSTRACT

BENIFITS

- Efficient Resource Management: LMS automates various library operations such as cataloging, circulation,efficient resource management. Librarians can easily track items, their availability.
- Enhanced Accessibility: LMS enables users to access library resources remotely, through online catalogs and digital collections. This enhances accessibility, for users who cannot physically visit the library.
- Integration with Digital Resources: Modern LMSs often integrate with digital repositories and electronic resources, allowing libraries to manage both physical and digital collections seamlessly.
- Cost Savings: While implementing an LMS requires an initial investment, it can lead to long-term cost savings through improved efficiency, reduced paperwork, and better resource utilization.

PROJECT FOCUS

This project aims to develop a user-friendly, secure, and efficient library management system using Python for front end development and MySQL as the relational database management system.

The project mainly focuses to create the LMS with :

- User-centric design
- Open source development
- High security and privacy
- Scalability and customisation

PROJECT FOCUS

The project will focus on core functionalities like:

- User registration
- List of books
- Books available
- Due date
- Details Of Author
- Issue Date
- Number Of Copies

IMPLEMENTATION

FRONT-END

```
from tkinter import*
from tkinter import ttk
from tkinter import messagebox
from PIL import Image,ImageTk
import random
import sqlite3
image1='library.png'
image2='image2.png'
image3='finance.png'
#pip install Pillow
class menu:

 def __init__(self):
  self.root=Tk()
  self.root.title('Menu')
  self.root.state('zoomed')
  conn=sqlite3.connect('test.db')
  conn.execute('''create table if not exists book_info
  (ID VARCHAR PRIMARY KEY NOT NULL,
  TITLE VARTEXT NOT NULL,
  AUTHOR VARTEXT NOT NULL,
  GENRE VARTEXT NOT NULL,
  COPIES VARINT NOT NULL,
  LOCATION VARCHAR NOT NULL);''')
  conn.commit()
```

```
conn.execute('''create table if not exists book_issued
  (BOOK_ID VARCHAR NOT NULL,
  STUDENT_ID VARCHAR NOT NULL,
  ISSUE_DATE DATE NOT NULL,
  RETURN_DATE DATE NOT NULL,
  PRIMARY KEY (BOOK_ID,STUDENT_ID));''')
  conn.commit()
  conn.close()
  self.a=self.canvases(image1)
   l1=Button(self.a,text='BOOK DATA',font='Papyrus
22
bold',fg='Yellow',bg='Black',width=19,padx=10,border
width=0,command=self.book).place(x=100,y=500)
                l2=Button(self.a,text='STUDENT
DATA',font='Papyrus                         22
bold',fg='Yellow',bg='Black',width=19,padx=10,border
width=0,command=self.student).place(x=800,y=500)
  self.root.mainloop()
 def canvases(self,images):
  w = self.root.winfo_screenwidth()
  h = self.root.winfo_screenheight()
  #photo=PhotoImage(file=images)
  photo=Image.open(images)
  photo1=photo.resize((w,h),Image.ANTIALIAS)
  photo2=ImageTk.PhotoImage(photo1)
```

FRONT-END

```
#photo2 = ImageTk.PhotoImage(Image.open(images).resize((w, h)),Image.ANTIALIAS)
  self.canvas = Canvas(self.root, width='%d'%w, height='%d'%h)
  self.canvas.grid(row = 0, column = 0)
  self.canvas.grid_propagate(0)
  self.canvas.create_image(0, 0, anchor = NW, image=photo2)
  self.canvas.image=photo2
  return self.canvas
 def book(self):
  self.a.destroy()
  self.a=self.canvases(image2)
  l1=Button(self.a,text='Add Books',font='Papyrus 22 bold',fg='Orange',bg='Black',width=15,padx=10,command=self.addbook).place(x=12,y=100)
  l2=Button(self.a,text='Search Books',font='Papyrus 22 bold',fg='Orange',bg='Black',width=15,padx=10,command=self.search).place(x=12,y=200)

  l4=Button(self.a,text='All Books',font='Papyrus 22 bold',fg='Orange',bg='Black',width=15,padx=10,command=self.all).place(x=12,y=300)
  l4=Button(self.a,text='<< Main Menu',font='Papyrus 22 bold',fg='Orange',bg='Black',width=15,padx=10,c

 def rm(self):
  self.f1.destroy()
 def mainmenu(self):
  self.root.destroy()
  a=menu()
```

FRONT-END

```
def addbook(self):
    self.aid=StringVar()
    self.aauthor=StringVar()
    self.aname=StringVar()
    self.acopies=IntVar()
    self.agenre=StringVar()
    self.aloc=StringVar()
    self.f1=Frame(self.a,height=500,width=650,bg='black')
    self.f1.place(x=500,y=100)
    l1=Label(self.f1,text='Book ID : ',font='Papyrus 12 bold',fg='Orange',bg='Black',pady=1).place(x=50,y=50)
    e1=Entry(self.f1,width=45,bg='orange',fg='black',textvariable=self.aid).place(x=150,y=50)
    l2=Label(self.f1,text='Title : ',font='Papyrus 12 bold',fg='Orange',bg='Black',pady=1).place(x=50,y=100)
    e2=Entry(self.f1,width=45,bg='orange',fg='black',textvariable=self.aname).place(x=150,y=100)
    l3=Label(self.f1,text='Author : ',font='Papyrus 12 bold',fg='orange',bg='Black',pady=1).place(x=50,y=150)
    e3=Entry(self.f1,width=45,bg='orange',fg='black',textvariable=self.aauthor).place(x=150,y=150)
    l4=Label(self.f1,text='Genre : ',font='Papyrus 12 bold',fg='orange',bg='Black',pady=1).place(x=50,y=200)
    e2=Entry(self.f1,width=45,bg='orange',fg='black',textvariable=self.agenre).place(x=150,y=200)
    l4=Label(self.f1,text='Copies : ',font='Papyrus 12 bold',fg='orange',bg='Black',pady=1).place(x=50,y=250)
    e2=Entry(self.f1,width=45,bg='orange',fg='black',textvariable=self.acopies).place(x=150,y=250)
    l5=Label(self.f1,text='Location : ',font='Papyrus 12 bold',fg='orange',bg='Black',pady=1).place(x=50,y=300)
    e3=Entry(self.f1,width=45,bg='orange',fg='black',textvariable=self.aloc).place(x=150,y=300)
    self.f1.grid_propagate(0)
    b1=Button(self.f1,text='Add',font='Papyrus 10 bold',fg='black',bg='orange',width=15,bd=3,command=self.adddata).place(x=150,y=400)
    b2=Button(self.f1,text='Back',font='Papyrus 10 bold',fg='black',bg='orange',width=15,bd=3,command=self.rm).place(x=350,y=400)
```

FRONT-END

```
def serch1(self):
    k=self.sid.get()
    if k!="":
      self.list4=("BOOK ID","TITLE","AUTHOR","GENRE","COPIES","LOCATION")
      self.trees=self.create_tree(self.f1,self.list4)
      self.trees.place(x=25,y=150)
      conn=sqlite3.connect('test.db')
        c=conn.execute("select * from book_info where ID=? OR TITLE=? OR AUTHOR=? OR GENRE=?",
(k.capitalize(),k.capitalize(),k.capitalize(),k.capitalize(),))
      a=c.fetchall()
      if len(a)!=0:
        for row in a:
        self.trees.insert("",END,values=row)
        conn.commit()
        conn.close()
        self.trees.bind('<<TreeviewSelect>>')
        self.variable = StringVar(self.f1)
        self.variable.set("Select Action:")
        self.cm =ttk.Combobox(self.f1,textvariable=self.variable ,state='readonly',font='Papyrus 15 bold',height=50,width=15,)
        self.cm.config(values =('Add Copies', 'Delete Copies', 'Delete Book'))
       self.cm.place(x=50,y=100)
        self.cm.pack_propagate(0)
        self.cm.bind("<<ComboboxSelected>>",self.combo)
        self.cm.selection_clear()
      else:
        messagebox.showinfo("Error","Data not found")
```

FRONT-END

```
d#=============================VARIABLES====================================
USERNAME = StringVar()
PASSWORD = StringVar()
#============================FRAMES======================================
'''Top = Frame(root, bd=2, relief=RIDGE)
Top.pack(side=TOP, fill=X)
Form = Frame(root, height=200)
Form.pack(side=BOTTOM, pady=20)'''
#============================LABELS======================================
lbl_title = Label(canvas, text = "ADMIN LOGIN", font=('Papyrus', 30,'bold', ),bg='black', fg='orange')
lbl_title.place(x=500,y=100)
lbl_username = Label(canvas, text = "Username:", font=('Papyrus', 15,'bold'),bd=4,bg='black', fg='orange')
lbl_username.place(x=500,y=230)
lbl_password = Label(canvas, text = "Password :", font=('Papyrus', 15,'bold'),bd=3, bg='black', fg='orange')
lbl_password.place(x=500, y=330)
lbl_text = Label(canvas)
lbl_text.place(x=450,y=500)
lbl_text.grid_propagate(0)
#============================ENTRY WIDGETS================================
username = Entry(canvas, textvariable=USERNAME, font=(14), bg='black', fg='orange',bd=6)
username.place(x=650, y=230,)
password = Entry(canvas, textvariable=PASSWORD, show="*", font=(14),bg='black', fg='orange',bd=6)
password.place(x=650, y=330)
#============================BUTTON WIDGETS===============================
btn_login = Button(canvas, text="LOGIN", font=('Papyrus 15 bold'),width=25,command=Login, bg='black', fg='orange')
btn_login.place(x=500,y=400)
btn_login.bind('<Return>', Login)
root.mainloop()
```

THANK YOU