Ministry of Higher Education
Benha Univeristy
Shoubra Faculty of Engineering
Electrical Engineering Department
Computer Systems Engineering Program

2022

3rd Year - 2nd Semester

# Database Design for a Library Management System

## Students:

Mahmoud Kamal Aldeen Mahmoud - 2166120180101166
Mohamed Khaled Mohamed - 2166120180101280
Mahmoud Musaad Abdelrahman - 2166120180100813
Mustafa Muhammed Hassan - 2166120180100845
Youssef Hassan Abdelhamed - 2166120180100965
Nihal Khaled Mohamed - 2166120180100902
Mariam Tarek Abdelghany - 2166120180100823
Mona Tarek Abdelghany - 2166120180100871
Doha Khaled Ahmed - 2166120180100408

Supervisor:
Dr. Sabah Saad Abdel Aziz

Database Design [ ECE323C ]

Benha University
(2021-2022)

# ACKNOWLEDGMENTS

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout our project to be completed successfully. we would like to express our deep and sincere gratitude to our project supervisor, Dr sabah for giving us the opportunity to work on this project and having such an experience in the database field and providing invaluable guidance throughout this project. We would like to express our thanks to Shimaa yousry and Mahmoud taha for their support and keen interest shown to complete this project. Special thanks to Draw.io website used for drawing the diagrams.

# ABSTRACT

Library management system is a system aims for developing a system to preserve the daily work of libraries using computers. This system maintains the information about the books available in the library, their authors, library customers and library staff. Maintenance of this information manually is a complex task. The online library management is designed to automate and computerize the operations performed over the information about the members, book issues and returns and all other operations. The workload of management is reduced as most of the manual work done is reduced. Computerized library has many features like facility of user and admin login. Admin login has more features in monitoring and managing the system.

# LIST OF FIGURES

# TABLE OF CONTENTS

# CHAPTER I: INTRODUCTION

A library management system is a collection of organized information or resources that is made accessible for borrowing or buying. These resources are supposed to be available in digital format or physical format (when the system is applied). The system will allow the users to store the book details. This system is supposed to be used by librarian to manage the library through a computerized system where he can add, sell, or delete books by using oracle. In non-computerized systems, sometimes there are loss of books, that won't happen in computerized systems.

## 1.1. Objectives:

The project aims and objectives are online book issue, Facility to download required book, admin login page where admin can add books, customer login page where student can find books, but the main aim is to develop a new automated system that will allow the librarians to maintain their daily work. It also keeps track of books' information such as cost, status or number of total books available

## 1.2 Literature Review :

### 1- LIBRARY MANAGEMENT SYSTEM USING PHP & MYSQL/MS ACCESS

Database management systems have become vital for organizations to manage large databases and to perform transactions upon such large data. These database applications not only store data, but also manage them, synchronize them, and help in information retrieval without errors. They reduce manual efforts and enhance the quality of information retrieval services. Due to this reason, they are widely used in almost all sectors. Libraries are popular places where there are numerous books to keep track of. Not only books but the librarian is also required to keep track of users, books that were taken, due dates, etc. Making manual entries and keeping track of due dates is not easy when the user's size is more. Thus, this work implements a library management system database application that helps the librarian manage all tasks in an efficient and user-friendly manner.

- **ER diagram for the Library Management System**

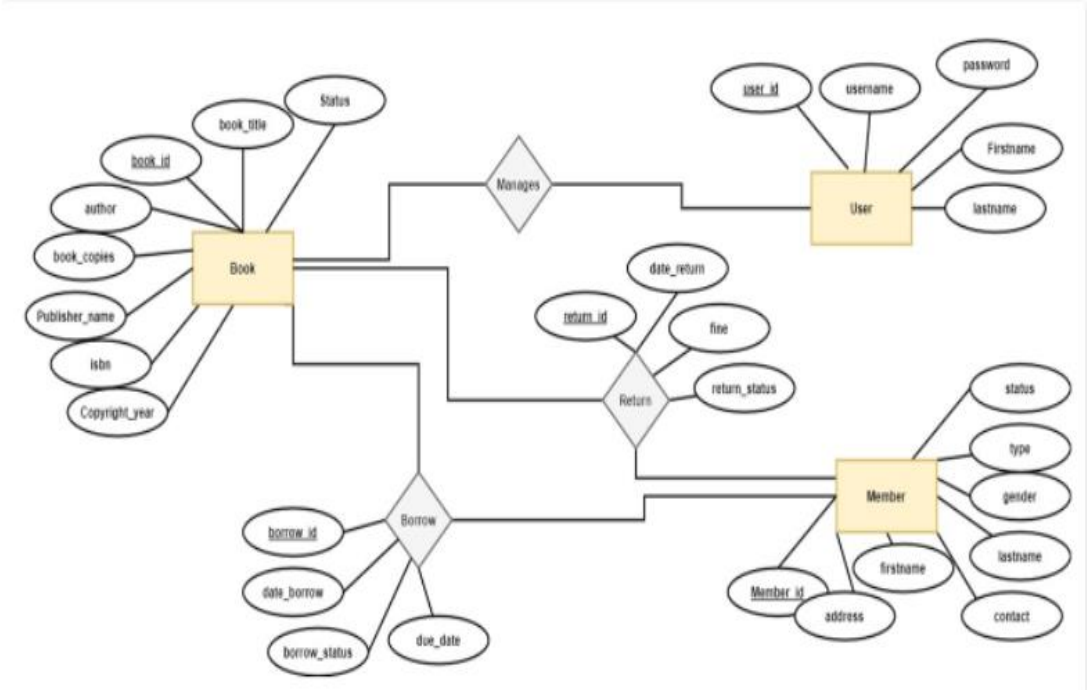



Fig (1): ER diagram for the Library Management System

- **Activity diagram**

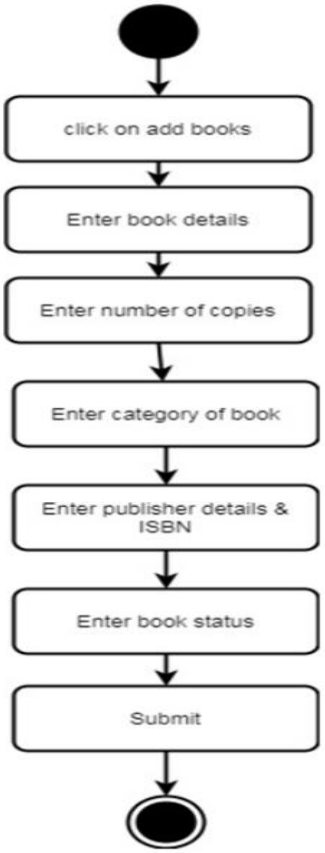
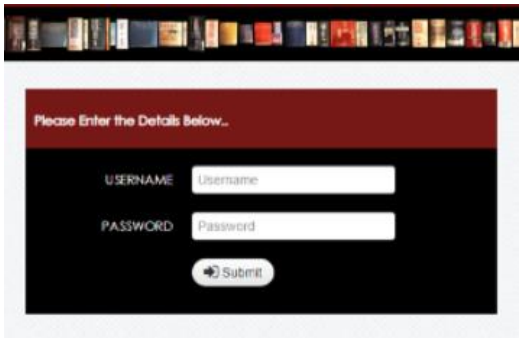



Fig (2): Activity Diagram

- **PROTOTYPE**



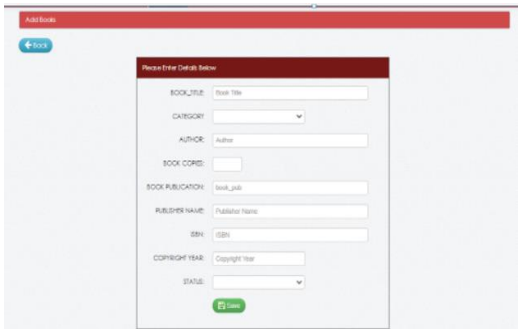



Fig (3): Prototype

## 2- Library Management System

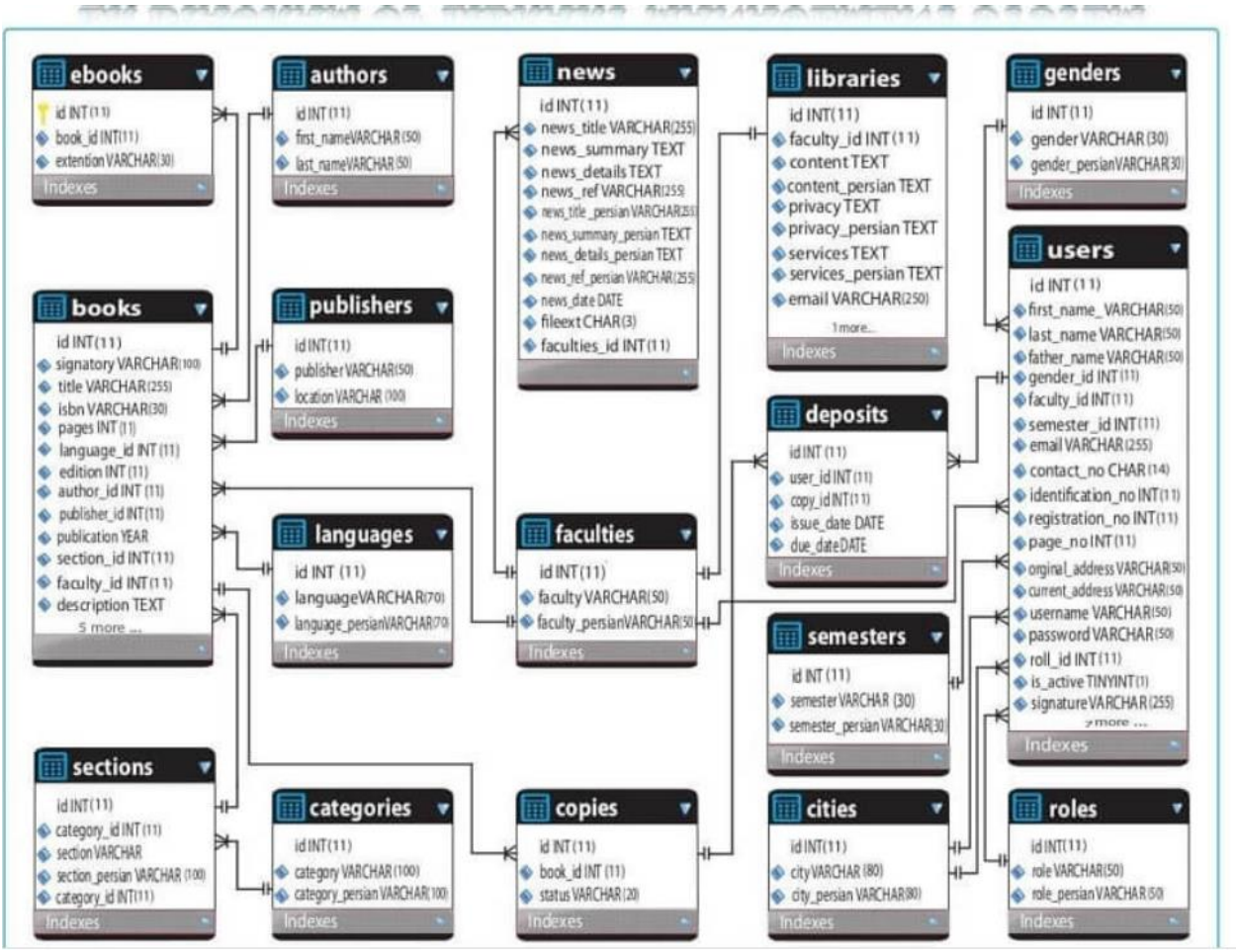
The library management system is an extensive database; the entity-relationship diagram helps show the relations between different entities and attributes. This helps in streamlining the things that are required for a particular Library management system.

To better understand an ER diagram, we will discuss a few examples and how they are used for the library management system.

As it can be observed in this example, the entities are present with their attributes. At the same time, the direct relations between the attributes are also shown, such as several CDs and books are present, so to differentiate, they have specific ISBNs and ISSNs. One key element to understand here is that proper symbols are used to understand an entity diagram.

- **ER DIAGRAM**

Fig (4): ER diagram of Library Management System example



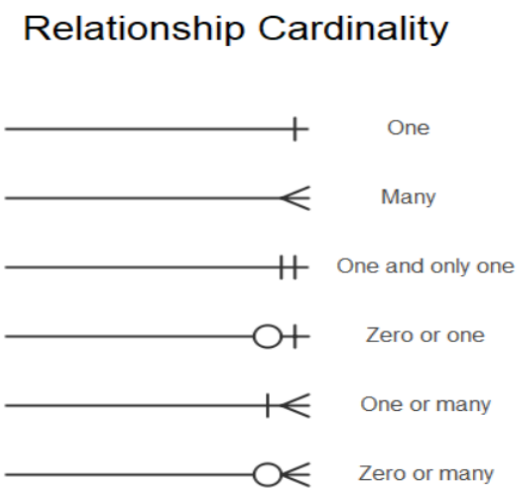



Fig (5): Relationship cardinality

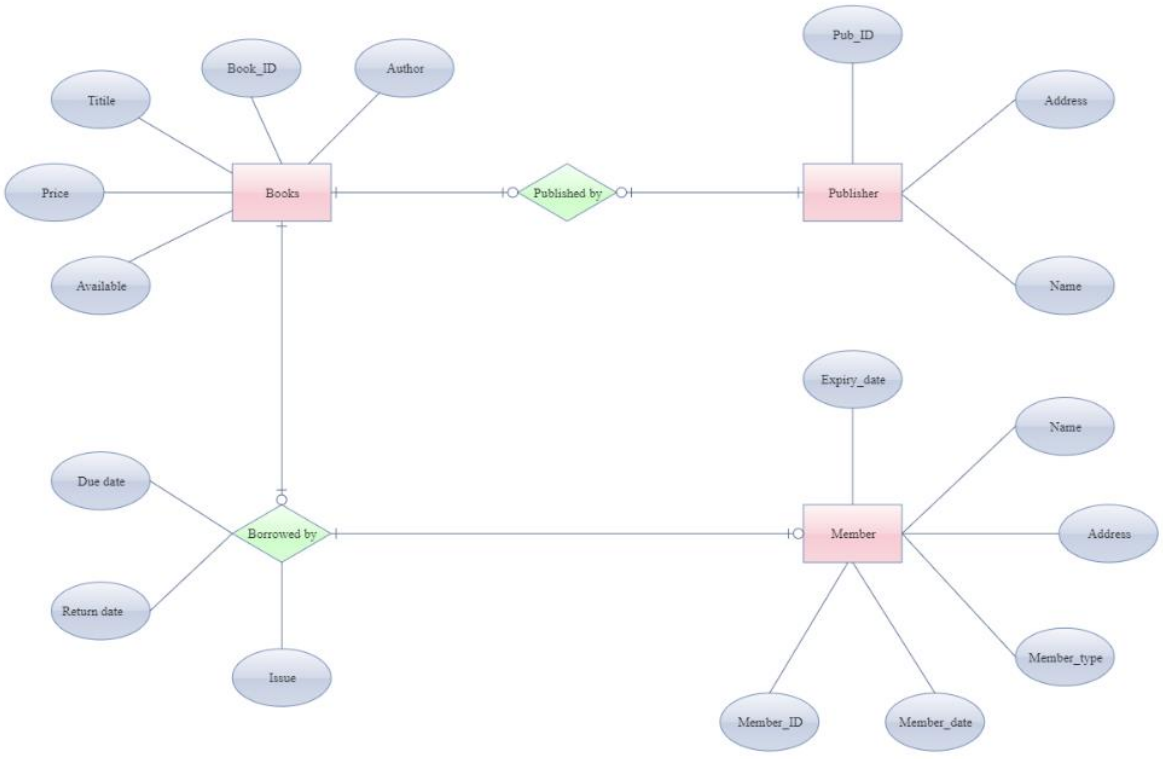



Fig (6): ERD shows the relationship cardinality

## 1.4. Contribution

*The design process consists of the following steps:*

- Determining the goal of the database
- Finding and organizing the information required
- Dividing the information into tables
- Turning information items into columns
- Specifying the primary keys
- Setting up the table relationships (ERD, ERRD, Schema diagram)
- Implementing the code using Oracle and Python
- Analyzing the design for errors

## 1.5 Outline

Chapter 2 of this report includes the details about materials and methods including software and hardware requirements. In chapter 3 design and implementation is discussed. Chapter 4 includes results and discussion and chapter 5 concludes the report.

# CHAPTER 2: Materials and Methods

## 2.1 RDBMS

A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database. RDBMS store data in the form of tables, with most commercial relational database management systems using Structured Query Language (SQL) to access the database. RDBMS is a program that allows you to create, update, and administer a relational database. The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation. An RDBMS is a type of database management system (DBMS) that stores data in a row-based table structure which connects related data elements. An RDBMS includes functions that maintain the security, accuracy, integrity and consistency of the data. This is different than the file storage used in a DBMS. The most common means of data access for the RDBMS is SQL.

*Uses of RDBMS:*

Relational database management systems are frequently used in disciplines such as manufacturing, human resources and banking. The system is also useful for airlines that need to store ticket service and passenger documentation information as well as universities maintaining student databases.

Some examples of specific systems that use RDBMS include IBM, Oracle, MySQL, Microsoft SQLServer and PostgreSQL.

## 2.2 MySQL

Structured Query Language is abbreviated as SQL which is a programming language used to communicate with data stored in a relational database management system. SQL syntax is similar to the English language, which makes it relatively easy to write, read, and interpret. Many RDBMSs use SQL to access the data in tables.

MySQL is the most popular open-source SQL database. It is typically used for web application development, and often accessed using PHP. MySQL is the most popular open-source SQL database. It is typically used for web application development, and often accessed using PHP. The main advantages of MySQL are that it is easy to use, inexpensive, reliable. Some of the disadvantages are that it has been known to suffer from poor performance when scaling, open source development has lagged since Oracle has taken control of MySQL, and it does not include some advanced features that developers may be used to. MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability.

## 2.3 Oracle

Oracle Database (Oracle DB) is a relational database management system from Oracle Corporation. Oracle Database is a multi-model database management system produced and marketed by Oracle Corporation and it is a collection of data treated as a unit. The system is built around a relational database framework in which data objects may be directly accessed by users through structured query language. Oracle is a fully scalable relational database architecture and is often used by global enterprises which manage and process data across wide and local area networks. The Oracle database has its own network component to allow communications across networks.

Fig (7): Oracle DB


Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost-effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

It is a database commonly used for running online transaction processing, data warehousing and mixed database workloads. Oracle Database is available by several service providers on-prem, on-cloud, or as hybrid cloud installation. It may be run on third party servers as well as on Oracle hardware.

# CHAPTER 3: Design and Implementation

### 3.1 ERD and EERD:

#### ERD

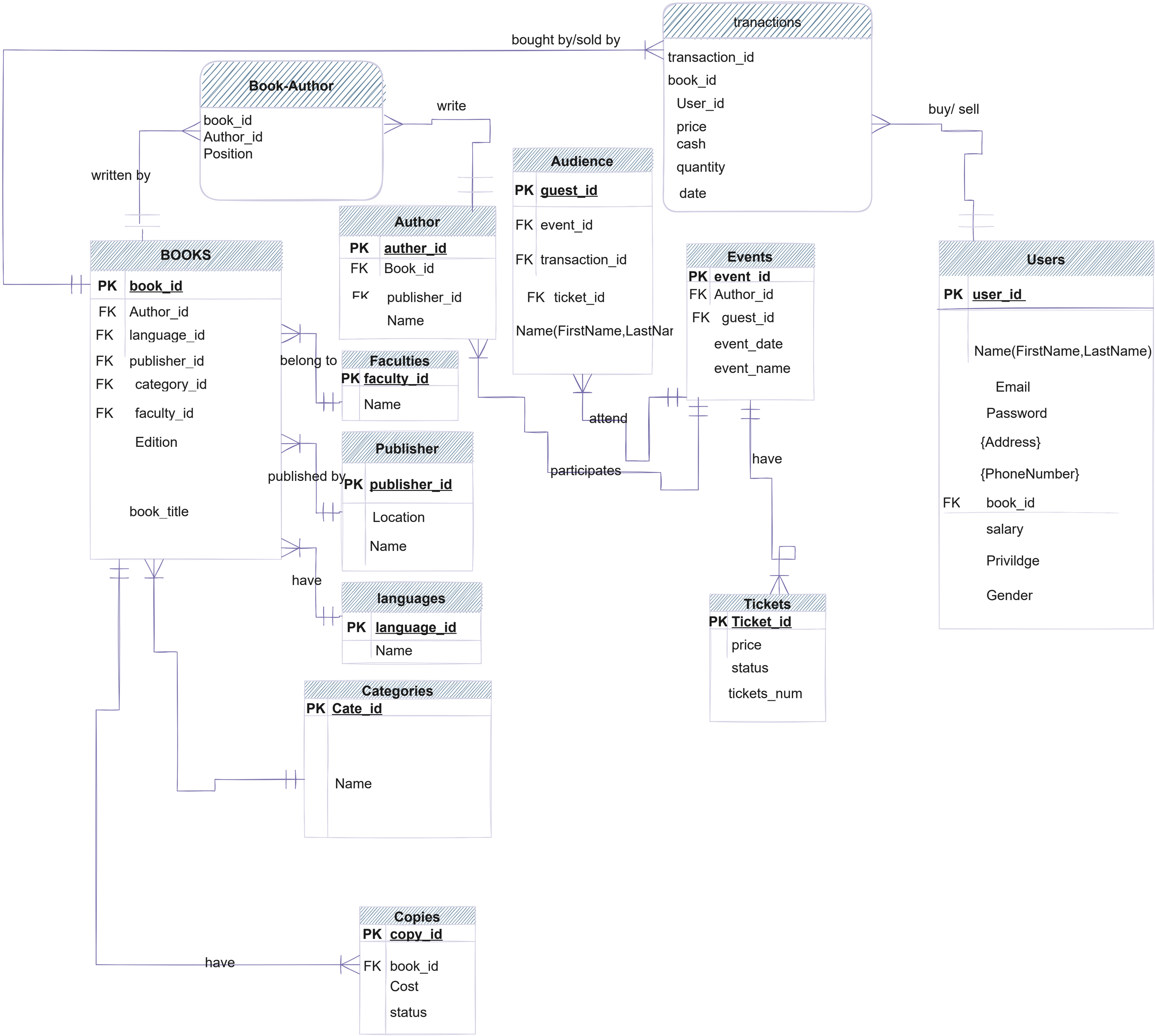
Entity relationship diagram is used in modern database software engineering to illustrate logical structure of database. It is a relational schema database modelling method used to model a system and approach and this approach commonly used in database design.

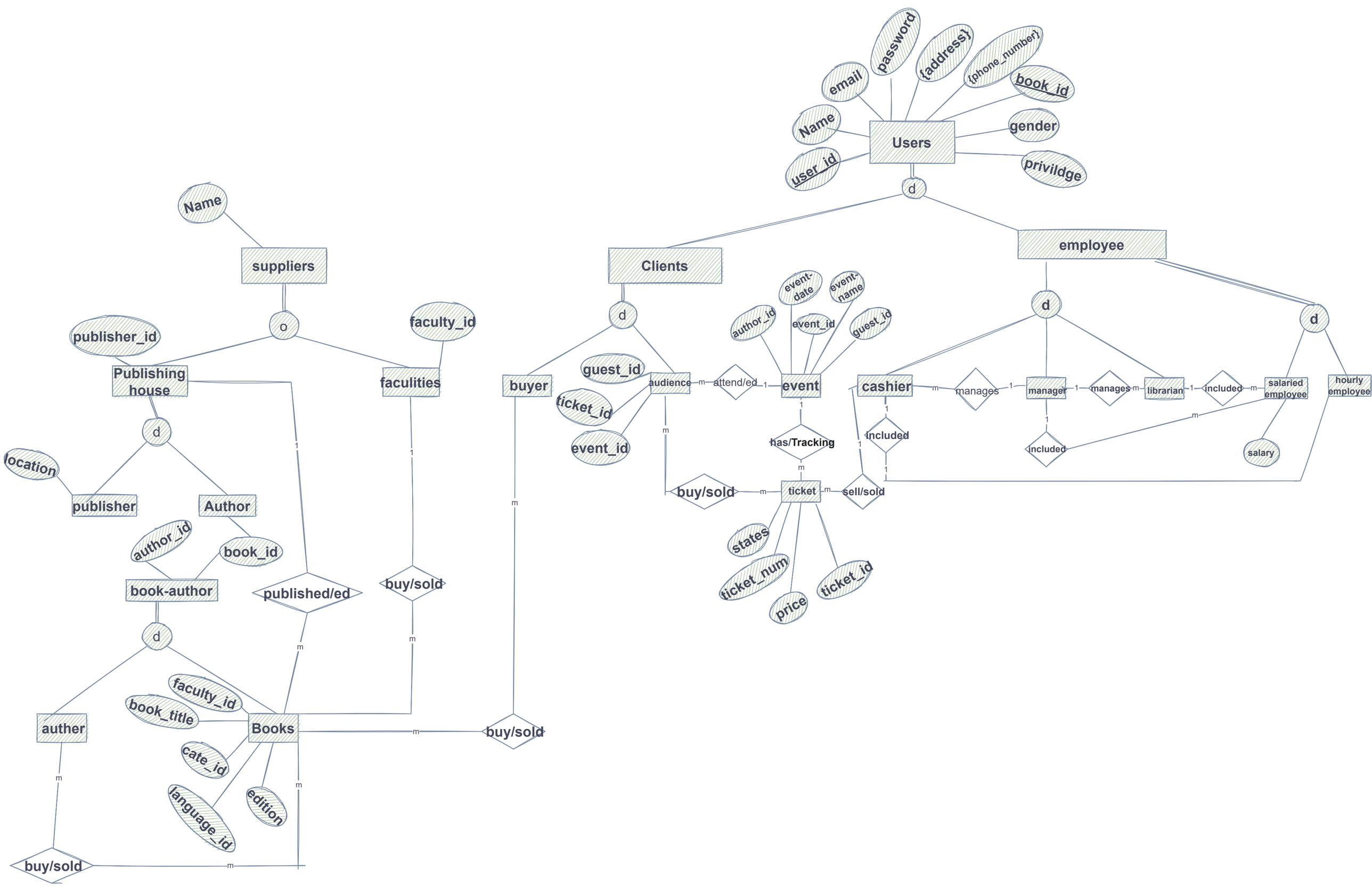
Uses of entity relationship diagrams:

Database design - Database troubleshooting - Business information systems - Business process re-engineering – Education – Research

#### EERD

An Enhanced entity–relationship diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER-diagram depicts the various relationships among entities, considering each object as entity. Entity is represented as rectangle shape and relationship represented as diamond shape. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.


tranactions
bought by/sold by
transaction_id
book_id
User_id
price
cash
quantity
date
Book-Author
book_id
Author_id
Position
write
written by
buy/ sell
Audience
PK guest_id
FK event_id
FK transaction_id
FK ticket_id
Name(FirstName,LastNar
Author
PK auther_id
FK Book_id
FK publisher_id
Name
Events
PK event_id
FK Author_id
FK guest_id
event_date
event_name
Users
PK user_id
Name(FirstName,LastName)
Email
Password
{Address}
{PhoneNumber}
FK book_id
salary
Privildge
Gender
BOOKS
PK book_id
FK Author_id
FK language_id
FK publisher_id
FK category_id
FK faculty_id
Edition
book_title
belong to
Faculties
PK faculty_id
Name
published by
Publisher
PK publisher_id
Location
Name
attend
participates
have
have
languages
PK language_id
Name
Tickets
PK Ticket_id
price
status
tickets_num
Categories
PK Cate_id
Name
have
Copies
PK copy_id
FK book_id
Cost
status


Name
suppliers
o
publisher_id
faculty_id
Publishing house
faculities
d
location
publisher
Author
author_id
book_id
book-author
d
published/ed
buy/sold
auther
faculty_id
book_title
Books
cate_id
language_id
edition
buy/sold
buy/sold
email
password
{address}
{phone_number}
book_id
Name
gender
user_id
privildge
Users
d
Clients
employee
d
d
buyer
guest_id
ticket_id
event_id
audience
attend/ed
author_id
event_id
event-date
event-name
guest_id
event
has/Tracking
buy/sold
ticket
sell/sold
states
ticket_num
price
ticket_id
cashier
manages
manager
manages
librarian
included
salaried employee
hourly employee
included
included
salary

**3.2 Schema Diagram:**

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema's underlying mission business rules for which the database is created. Database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagram. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram.

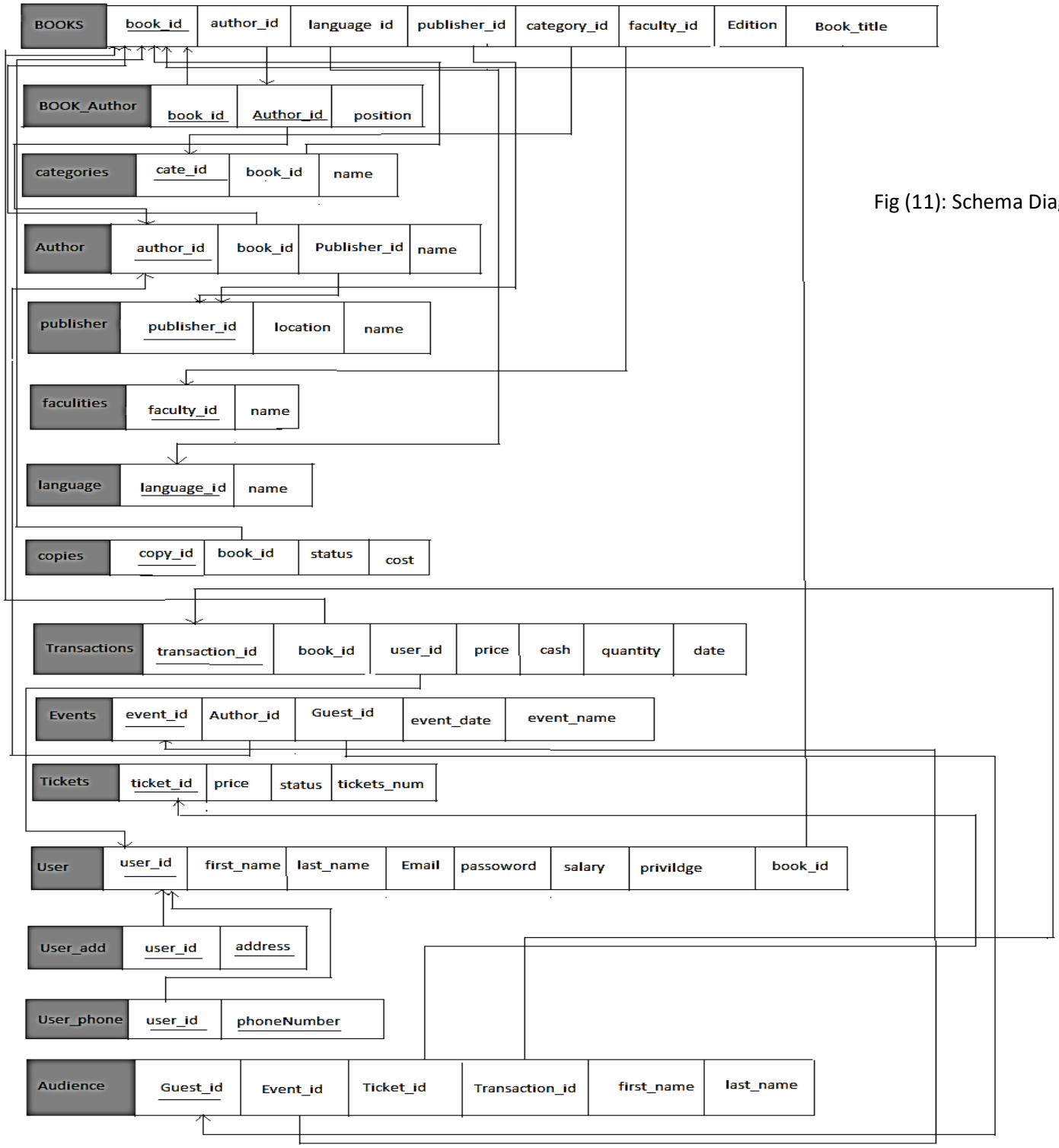
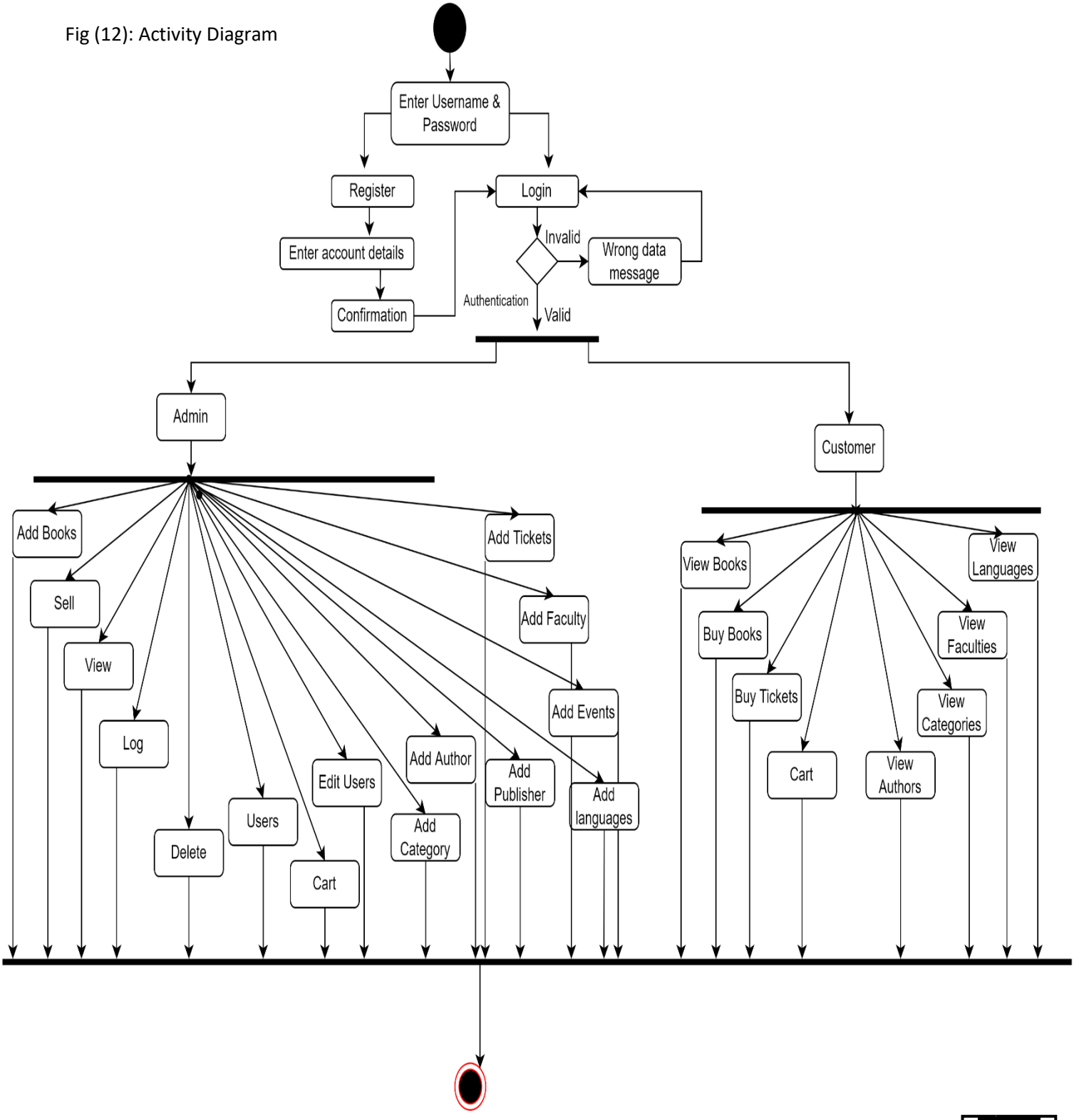



Fig (11): Schema Diagram

### 3.3: Activity Diagram

Activity diagrams is used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram. An activity diagram is a behavioral diagram. It depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

Fig (12): Activity Diagram



### 3.3 Implementation

All code files are uploaded to this GitHub repository:

https://github.com/MahmoudKamal01/Database-Design-for-a-Library-Management-System

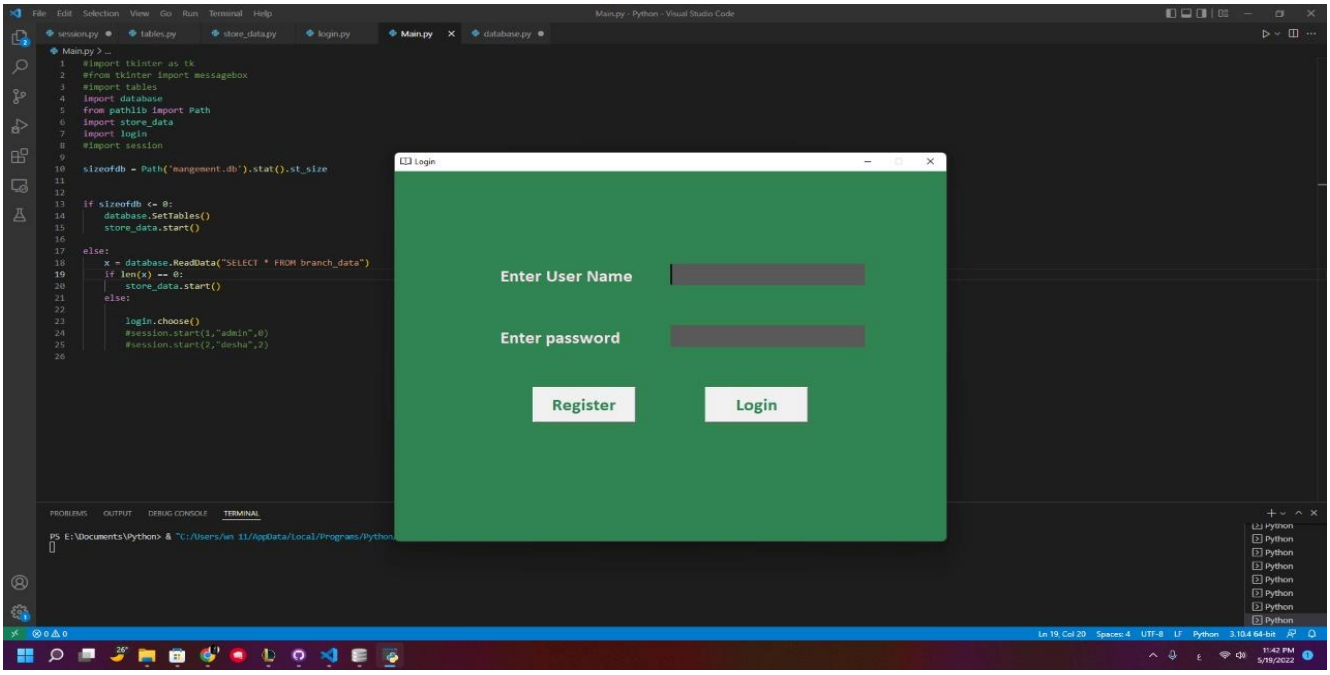



Fig (13): Login page

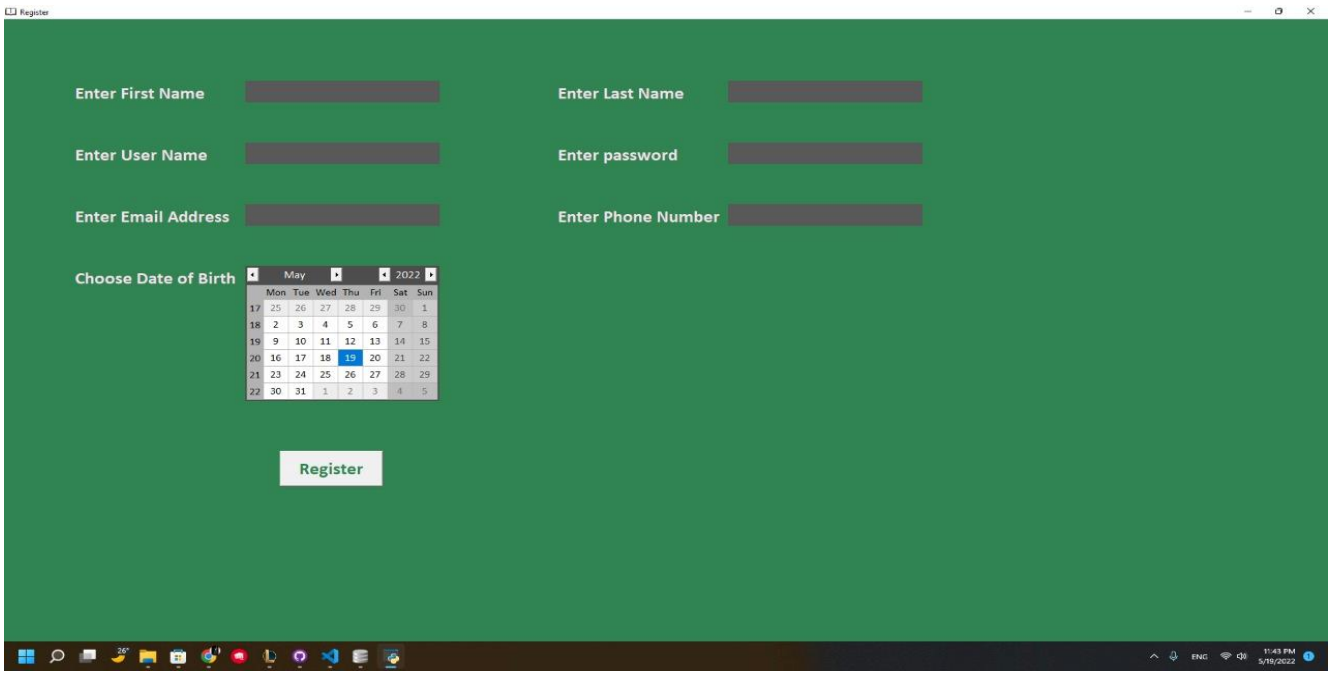



Fig (14): Register page

# CHAPTER 4: Results and Discussion

In this chapter results of the project are discussed.

## 4.1 White Box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. It is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing. In white box testing, code is visible to testers. We can see through box concept so its called white box.

The main goal of white box testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected outputs. When a specific input does not result in the expected output, then you have encountered a bug.

## 4.2 Test Cases

Log in test:

| Test case number | 1 |
|---|---|
| Input | Correct Username and Correct Password |
| Action | Press Log In Button |
| Expected Output | System should open main window for logged account |
| Actual Output | System opens main window |
| Decision | Normal, Successful Output |

| Test case number | 2 |
|---|---|
| Input | Correct Username and wrong Password |
| Action | Press Log in Button |
| Expected Output | System should display an error message |
| Actual Output | Wrong password |
| Decision | Normal, Successful Output |

Register Test:

| Test case number | 3 |
|---|---|
| Input | Admin username |
| Action | Press Register Button |
| Expected Output | System shows an error message |
| Actual Output | Can’t use “admin” as username |
| Decision | Normal, Successful Output |

| Test case number | 4 |
|---|---|
| Input | Unique username and password |
| Action | Press register Button |
| Expected Output | System goes to log in form |
| Actual Output | Successful |
| Decision | Normal, Successful Output |

Book Issue test:

| Test case number | 5 |
|---|---|
| Input | Filling fields with proper data about book |
| Action | Press add book button |
| Expected Output | Book added successfully |
| Actual Output | Book is now available |
| Decision | Normal, Successful Output |

| Test case number | 6 |
|---|---|
| Input | User enters the book name |
| Action | Press Delete Button |
| Expected Output | Book Removed successfully |
| Actual Output | Book is not available anymore |
| Decision | Normal, Successful Output |

## 4.3 Results

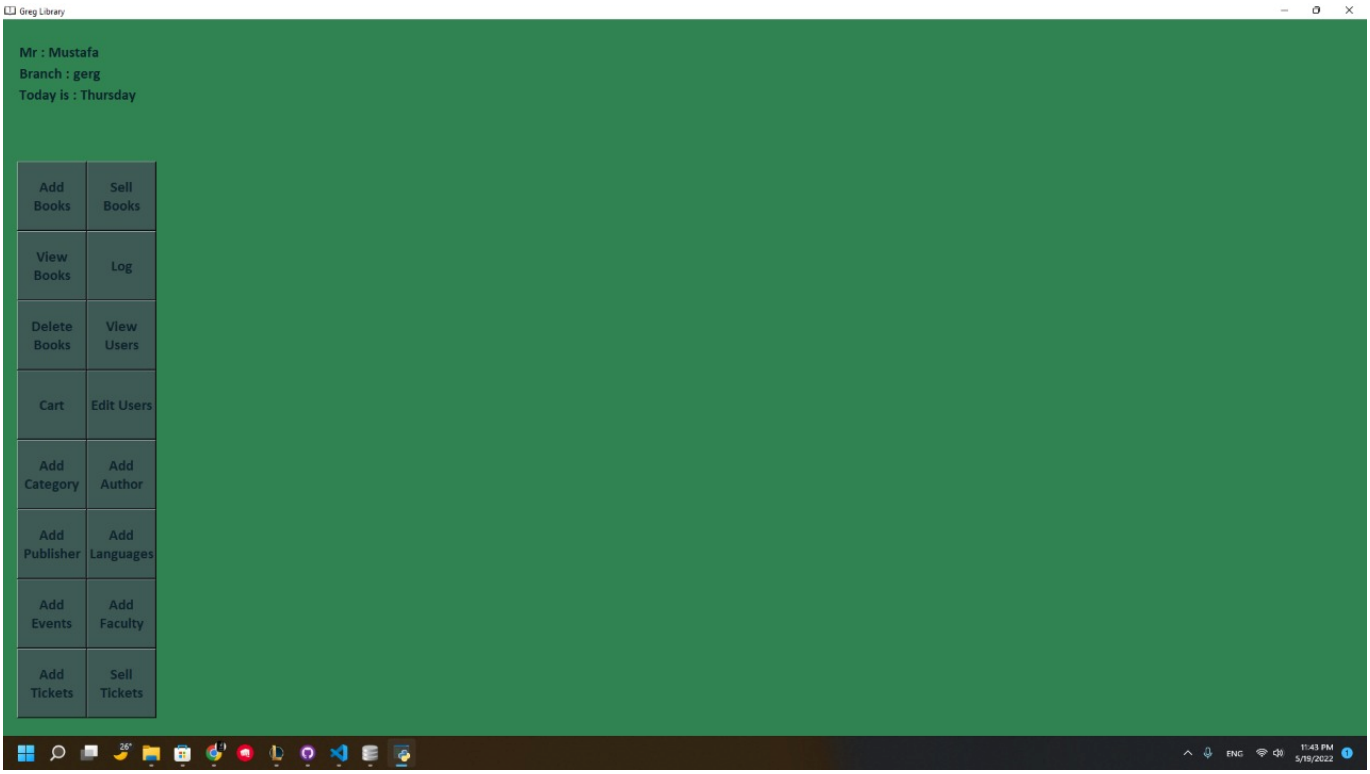



Fig (15): Admin page

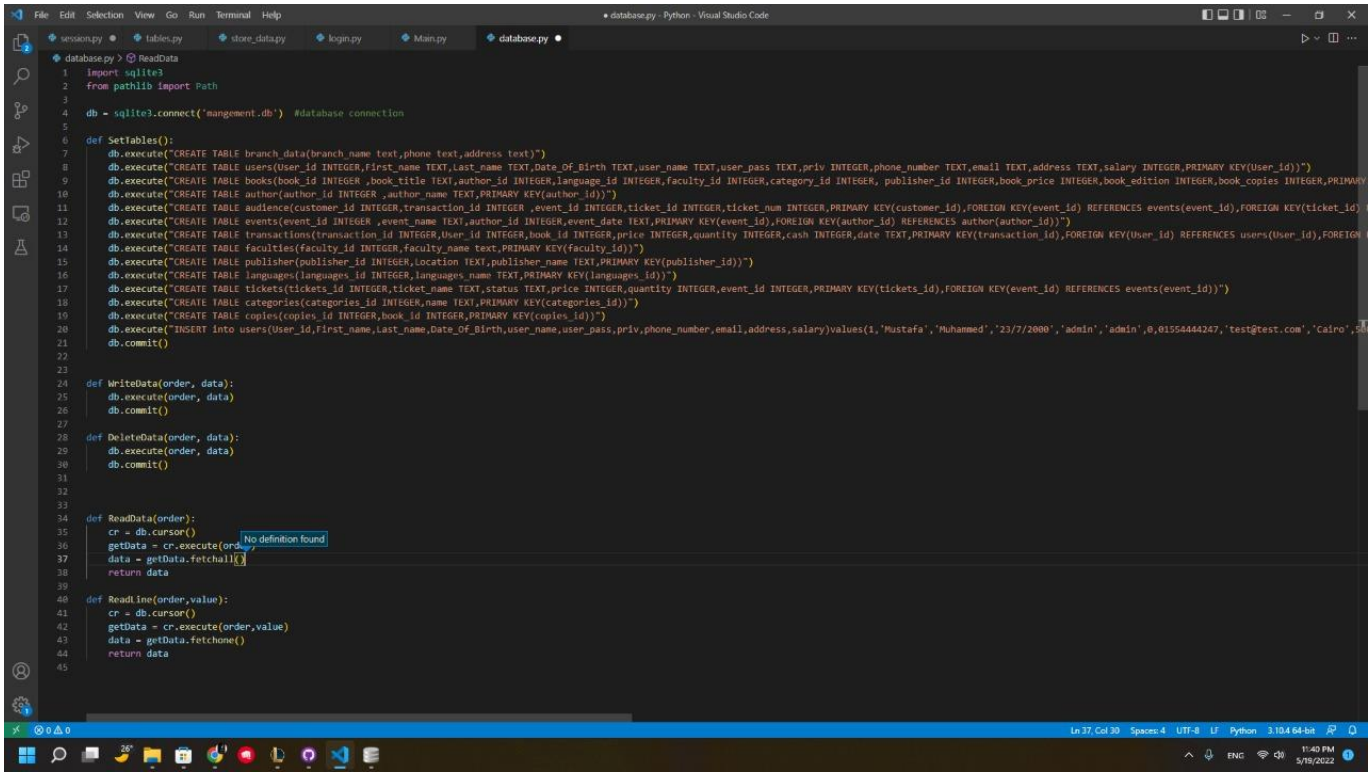



Fig (16): database.py file

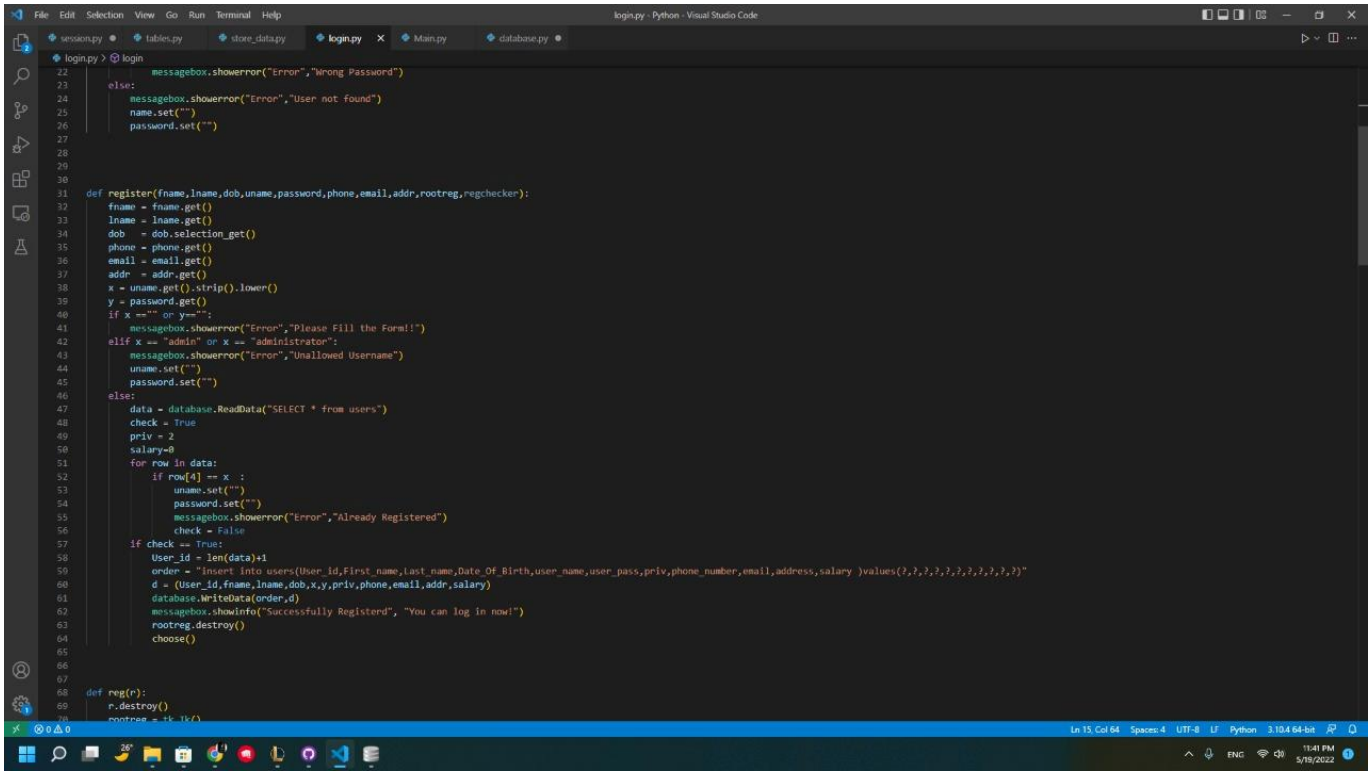
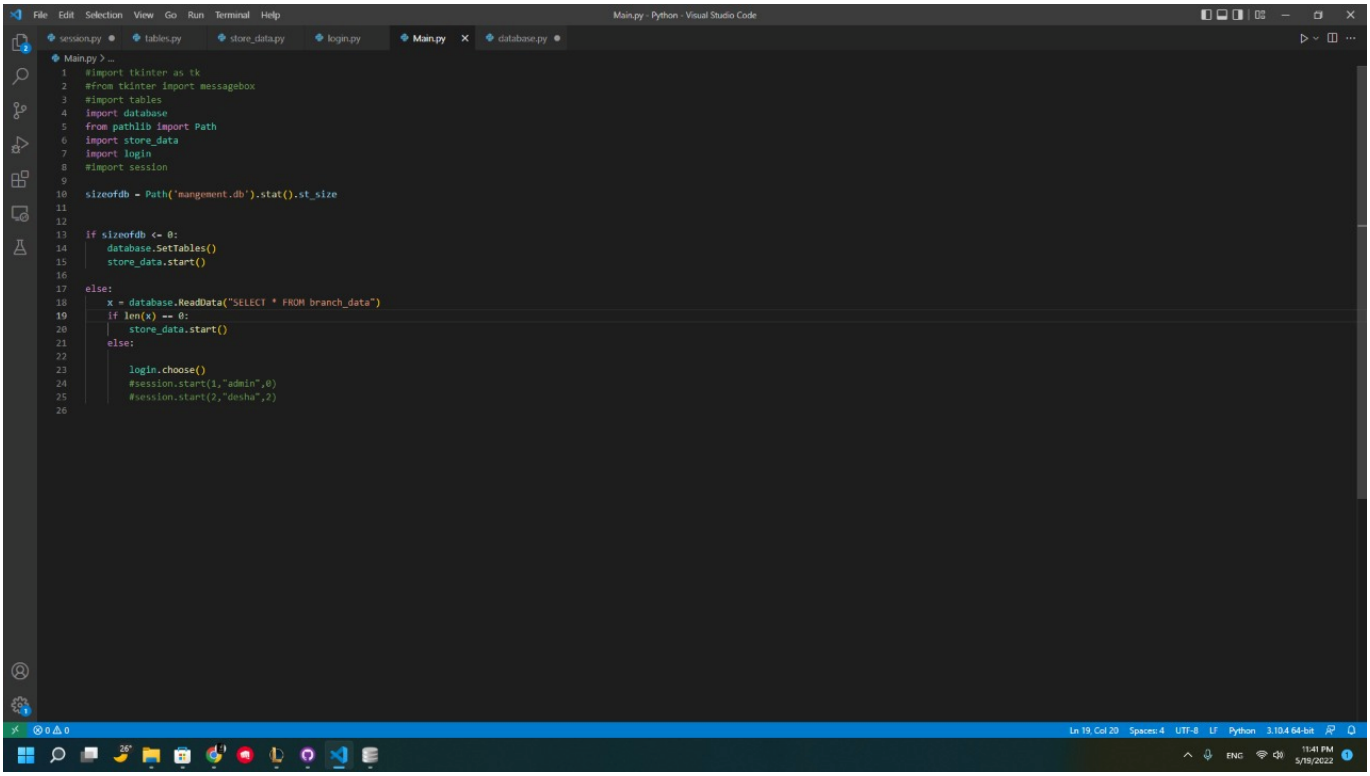
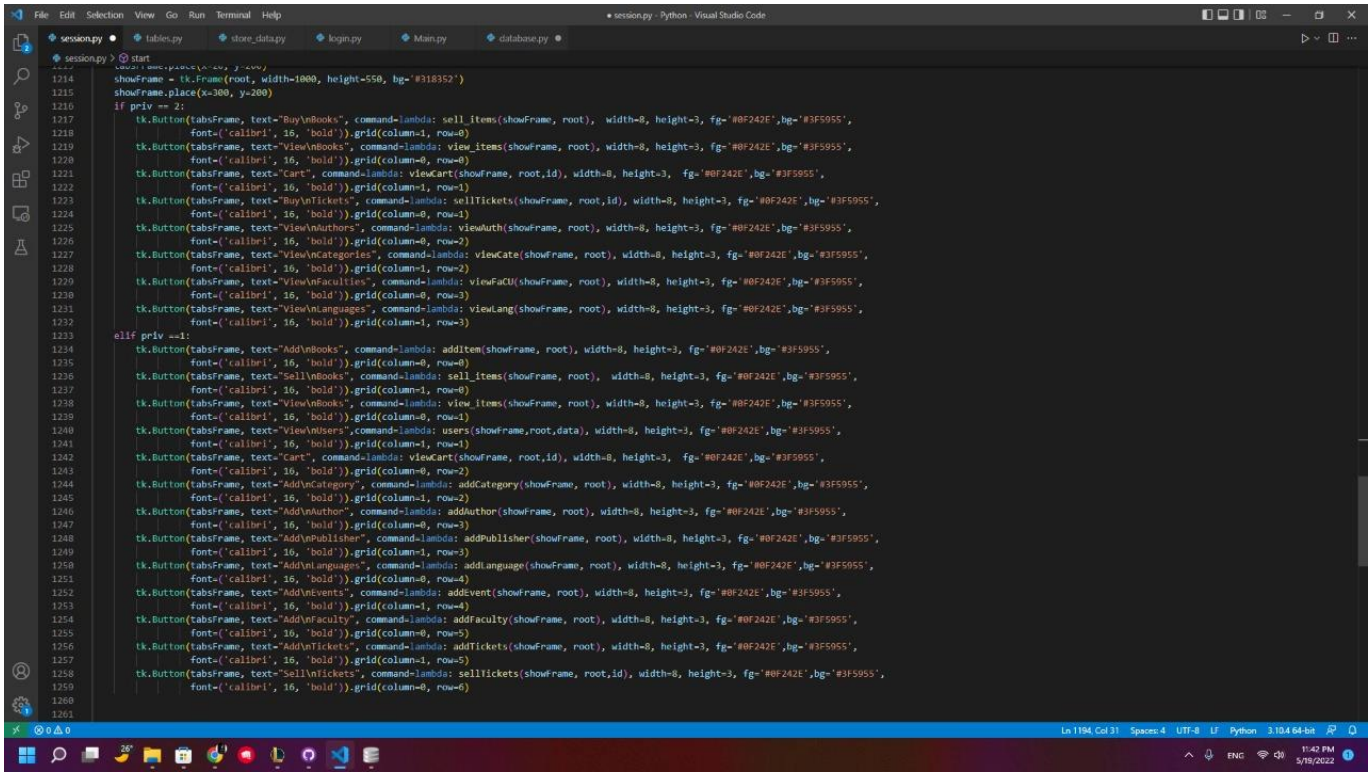



Fig (17): login.py file



```
#import tkinter as tk
#from tkinter import messagebox
#import tables
import database
from pathlib import Path
import store_data
import login
#import session

sizeofdb = Path('mangement.db').stat().st_size


if sizeofdb <= 0:
    database.SetTables()
    store_data.start()

else:
    x = database.ReadData("SELECT * FROM branch_data")
    if len(x) == 0:
        store_data.start()
    else:

        login.choose()
        #session.start(1,"admin",0)
        #session.start(2,"desha",2)
```

Fig (18): Main.py file



```
    showFrame = tk.Frame(root, width=1000, height=550, bg='#318352')
    showFrame.place(x=300, y=200)
    if priv == 2:
        tk.Button(tabsFrame, text="Buy\nBooks", command=lambda: sell_items(showFrame, root),  width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=0)
        tk.Button(tabsFrame, text="View\nBooks", command=lambda: view_items(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=0)
        tk.Button(tabsFrame, text="Cart", command=lambda: viewCart(showFrame, root,id), width=8, height=3,  fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=1)
        tk.Button(tabsFrame, text="Buy\nTickets", command=lambda: sellTickets(showFrame, root,id), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=1)
        tk.Button(tabsFrame, text="View\nAuthors", command=lambda: viewAuth(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=2)
        tk.Button(tabsFrame, text="View\nCategories", command=lambda: viewCate(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=2)
        tk.Button(tabsFrame, text="View\nFaculties", command=lambda: viewFaCU(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=3)
        tk.Button(tabsFrame, text="View\nLanguages", command=lambda: viewLang(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=3)
    elif priv ==1:
        tk.Button(tabsFrame, text="Add\nBooks", command=lambda: addItem(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=0)
        tk.Button(tabsFrame, text="Sell\nBooks", command=lambda: sell_items(showFrame, root),  width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=0)
        tk.Button(tabsFrame, text="View\nBooks", command=lambda: view_items(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=1)
        tk.Button(tabsFrame, text="View\nUsers",command=lambda: users(showFrame,root,data), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=1)
        tk.Button(tabsFrame, text="Cart", command=lambda: viewCart(showFrame, root,id), width=8, height=3,  fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=2)
        tk.Button(tabsFrame, text="Add\nCategory", command=lambda: addCategory(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=2)
        tk.Button(tabsFrame, text="Add\nAuthor", command=lambda: addAuthor(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=3)
        tk.Button(tabsFrame, text="Add\nPublisher", command=lambda: addPublisher(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=3)
        tk.Button(tabsFrame, text="Add\nLanguages", command=lambda: addLanguage(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=4)
        tk.Button(tabsFrame, text="Add\nEvents", command=lambda: addEvent(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=4)
        tk.Button(tabsFrame, text="Add\nFaculty", command=lambda: addFaculty(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=5)
        tk.Button(tabsFrame, text="Add\nTickets", command=lambda: addTickets(showFrame, root), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=1, row=5)
        tk.Button(tabsFrame, text="Sell\nTickets", command=lambda: sellTickets(showFrame, root,id), width=8, height=3, fg='#0F242E',bg='#3F5955',
                  font=('calibri', 16, 'bold')).grid(column=0, row=6)
```

Fig (19): session.py file

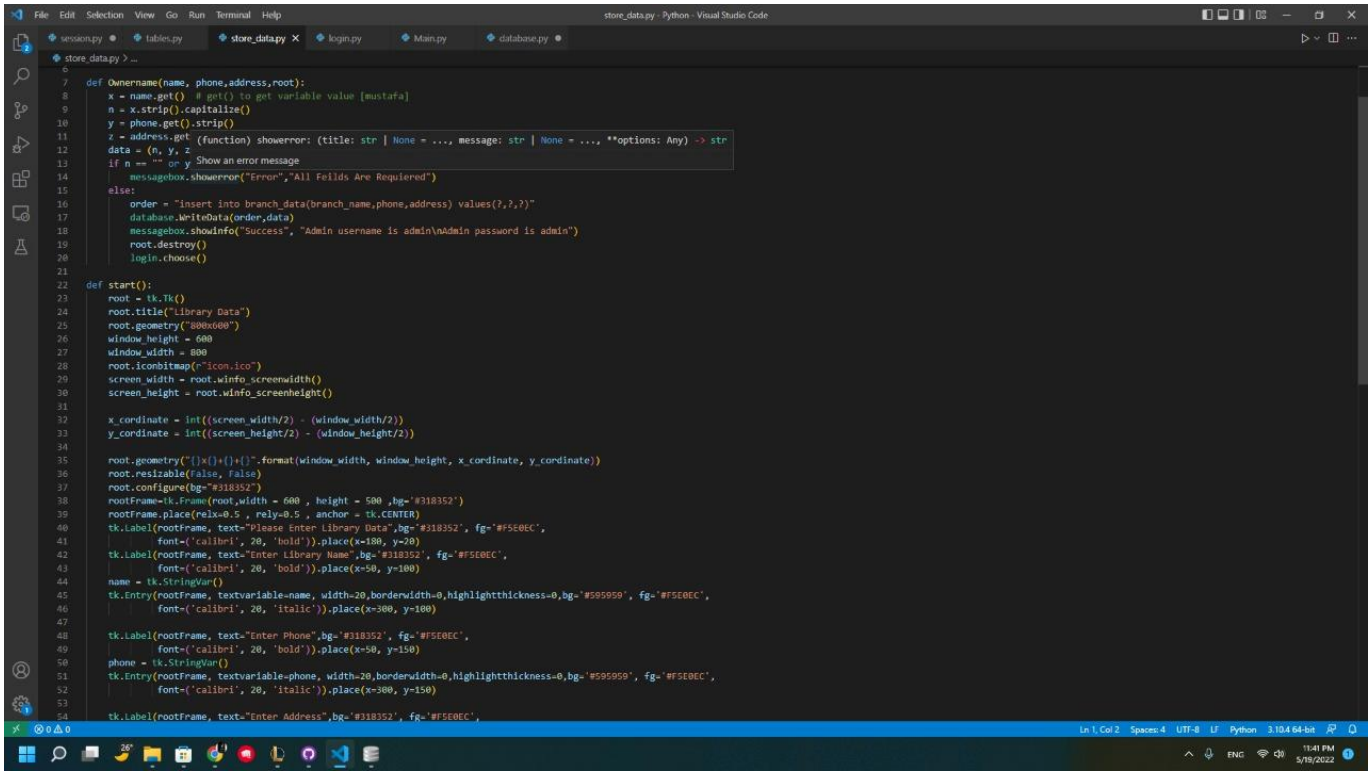



Fig (20): store_data.py file

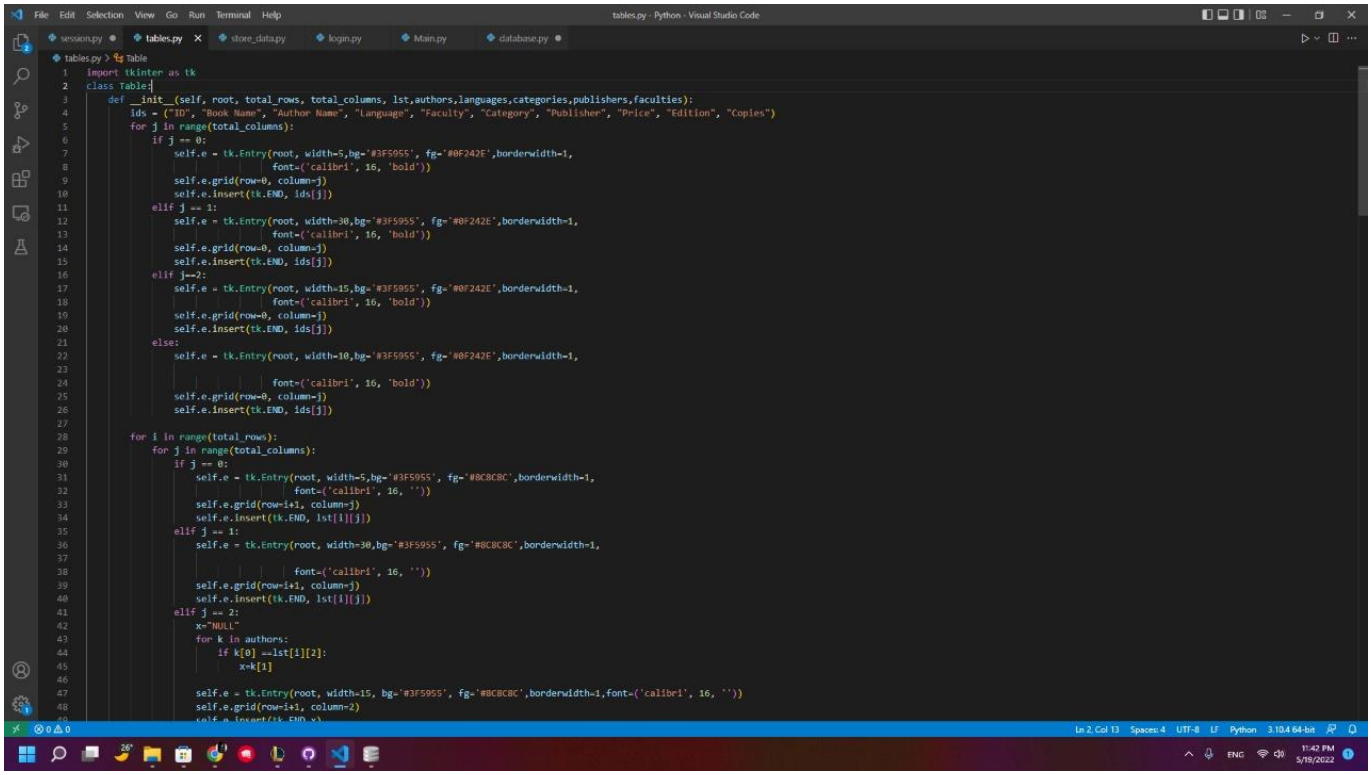



Fig (21): tables.py file

# CHAPTER 5: Summary and Future Work

## 5.1: Summary

After we have completed the project we are sure the problems in the existing system would be overcame. The “LIBRARY MANAGEMENT SYSTEM” process allows the user to store the book details and the person's details. This software allow storing the details of all the data related to library. The implementation of the system will reduce the data entry time and provide readily calculated reports.

### 5.2: Future Work

Our project can be improved by many strategies. Take your library everywhere you go and get real-time updates of circulation on iPhone, iPad and Android devices. Users can conveniently access the library collections from an application or a website so they can buy the book they need online. Librarian can schedule programs using events calendar and share with members. Another strategy that free e-books can be available for reading on library website or application.

**REFERENCES**


[1]- https://www.techopedia.com/definition/8711/oracle-database

[2]- https://www.oracletutorial.com/getting-started/what-is-oracle-database/

[3]- https://www.techopedia.com/definition/8711/oracle-database

[4]- https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm

[5]- https://www.codecademy.com/article/what-is-rdbms-sql

[6]- https://www.techtarget.com/searchdatamanagement/definition/RDBMS-relational-database-management-system#

[7]- https://www.lucidchart.com/pages/er-diagrams

[8]- https://afteracademy.com/blog/what-is-a-schema

[9]- https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams

[10]- https://www.guru99.com/white-box-testing.html