

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka - 590018



A MINI PROJECT REPORT

ON

LIBRARY MANAGEMENT SYSTEM

SUBMITTED BY:

NIKITH KUMAR N (1RI17CS028)

Under the guidance of

Prof. Mala P

Assistant Professor, Dept. of CSE



Department of Computer Science and Engineering

R.R. Institute of Technology

Bangalore, Karnataka, India – 560 090

November 2019 - 20

R R INSTITUTE OF TECHNOLOGY

No. 67, Raja Reddy layout, Hesaraghatta Main Road, Chikkabanavara, Bangalore - 560090

Department of Computer Science and Engineering



CERTIFICATE

Certified that the mini project work prescribed in 17CSL58 entitled “**LIBRARY MANAGEMENT SYSTEM**” carried out by **Mr.NIKITH KUMAR N (IRI17CS028)** a bonafied student of R R Institute Of Technology, Bengaluru in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2019 - 20. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said Degree.

.....
Signature of Internal Guide

[Prof.Mala P]

Assistant professor,
Dept.of CSE, RRIT

.....
Signature of HOD

[Dr.Manjunath R]

Prof. and Head of Dept,
Dept. of CSE, RRIT

.....
Signature of Principal

[Dr.Srinivas G Bhat]

Principal
RRIT, Bangalore

Name of the Examiners

1.....

2.....

Signature with date

.....

.....

ACKNOWLEDGEMENT

The completion of mini project work brings a sense of satisfaction, but it is never complete without thanking the persons responsible for its successful completion.

At the outset I express our most sincere grateful acknowledgment to the holy sanctum “**R R Institute of Technology**”, the temple of learning, for giving me an opportunity to pursue the degree course in Computer Science and Engineering and thus helping me in shaping my career.

I extend my deep sense of sincere gratitude to **Dr. Srinivas .G Bhat**, Principal, R R Institute of Technology, Bengaluru, for providing me an opportunity to continue my higher studies.

I express my heartfelt sincere gratitude to **Dr. Manjunath .R**, Professor and HOD, Department of Computer Science and Engineering, R R Institute of Technology, Bengaluru, for her valuable suggestions and support.

I extend my special in-depth, heartfelt, sincere gratitude to my guide **Prof. MALA .P**, Assistant Professor, Department of Computer Science and Engineering, R R Institute of Technology, Bengaluru, for her constant support and valuable guidance for completion of the mini project work.

I would like to thank all the teaching and non-teaching staff members in my Department of Computer Science and Engineering, R R Institute of Technology, Bengaluru, for their support.

Finally, I would like to thank all my friends and family members for their constant support, guidance and encouragement.

NIKITH KUMAR N

(IRI17CS028)

DECLARATION

I, **NIKITH KUMAR N**, student of 5th semester in Computer Science and Engineering, R R Institute of Technology, Bengaluru, hereby declare that the mini project entitled “**LIBRARY MANAGEMENT SYSTEM**” has been carried out by us under the supervision of our guide **Prof MALA.P**, Assistant Professor, Dept. of Computer Science and Engineering, R R Institute of Technology , Bengaluru and submitted in partial fulfillment for the award of degree in Bachelor of Engineering in Computer Science and Engineering of **Visvesvaraya Technological University, Belagavi** during the academic year 2019 - 2020. I further declare that the report has not been submitted to any other University for the award of any other degree.

NIKITH KUMAR N
(1RI17CS028)

Place: Bangalore

Date:

ABSTRACT

Library management system is a project which aims at developing a computerized system to maintain all the daily work of the library. This project has many features which are generally not available in normal library management system like facility of admin login through which the admin can monitor the whole system. The Admin after logging into his account can perform various operations such as adding information of new student, new faculties, new books, issue books, return book details, deleting books, deleting student details and so on.

And also Admin can delete the student details and book details. Students can borrow books from the library, and students can return their books before the due date. The time for returning the books is 7 days, if return date is exceeds then it calculates the fine per day 2Rs automatically.

The Library Management System software allows to Student to borrow only 3 books, if student borrowed more than 3 books, the system shows message ie; (This Student_id has taken 3 books, Try to return the books and borrow another one.)

The Library Management System is gaining more importance as the number of its users is increasing rapidly. As the number is rising there is a need of effective management of library, onesuch effective system is our Library Management System its designed using Tkinter GUI in Python as frontend and SQLITE3 as backend database.

The transactions like login, register, add, search, delete, issue, return are provided. The Library Management System stores the details like Student_ID, Name, Phone No, Semester, Department of the users who come to the library. The details of books like Book_ID, Book_Title, subject to which it belongs,author,edition, year of publication , the total price of books that are present in the library are stored. The details of faculties are Faculty_ID, Faculty_name, gender, department and contact no. that are present in the library are stored...

TABLE OF CONTENTS

CHAPTER NO.	CHAPTER NAME	PAGE NO.
1.	INTRODUCTION	01
	1.1 PROJECT AIMS AND OBJECTIVES	01
	1.2 BACKGROUND OF PROJECT	02
2.	REQUIREMENT ANALYSIS	03
	2.1 FUNCTIONS	03
	2.2 DATA	03
	2.3 SOFTWARE TOOLS USED	04
3.	SYSTEM ANALYSIS AND DESIGN	05
	3.1 SYSTEM ARCHITECTURE	05
	3.2 SCHEMA	06
	3.3 DATA FLOW DIAGRAMS	07
	3.3.1 DFD Level 0 of Initial Data	07
	3.3.2 DFD Level 1 of Main Page	08
	3.4 ACTIVITY DIAGRAM	09
	3.5 E R DIAGRAM	10
	3.6 NORMALIZATION	12
	3.6.1 First Normal Form	12
	3.6.2 Second Normal Form	13
	3.6.3 Third Normal Form	14
4.	RESULTS	15
	4.1 SNAPSHOTS	16

CONCLUSION AND FUTURE ENHANCEMENT	21
REFERENCES	22
APPENDIX	23
SOURCE CODE	23

LIST OF TABLES

TABLE NO.	TABLE NAMES	PAGENO
Fig 3.6.1	1st Normal Form	12
Fig 3.6.2	2nd Normal Form	13
Fig 3.6.3	3rd Normal Form	14

LIST OF FIGURES

FIGURE NO.	FIGURE NAMES	PAGENO
Fig 3.1	SYSTEM ARCHITECTURE	05
Fig 3.2	SCHEMA	06
Fig 3.3.1	DFD Level 0 of Initial Data	07
Fig 3.3.2	DFD Level 1 of Main Page	08
Fig 3.4	ACTIVITY DIAGRAM	09
Fig 3.5	E R DIAGRAM	11

CHAPTER - 1

INTRODUCTION

Library Management System software for monitoring and controlling the transactions in a library. The project “Library Management System” is developed in Python & Sqlite3 which mainly focuses on basic operations in a library like adding new student information, faculty information, books information, searching books and deleting books and student information and returning books.

1.1 PROJECT AIMS AND OBJECTIVES

The aim of this project is to develop a system that can handle and manage the activities involved in a library in an efficient and reliable way.

The objectives are:

The objective of the “Library Management Software” is to handle the entire activity of a library. The software keeps track of all the information about the books in the library, their cost, their complete details and total number of books available in the Library.

- The user will find it easy in this automated system rather than using the manual writing system. The system contains a database where all the information will be stored safely.
- The system is user-friendly and error free.
- Designing a computerized library management system which would help to save the cost and time.

Problem faced in manual library:

- Implementing the system.
- Evaluating and testing the performance of the system.

1.2 BACKGROUND OF PROJECT

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students, the addition of new faculties etc.

Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used.

In addition, report module is also included in Library Management System. If user's position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

CHAPTER - 2

REQUIREMENT ANALYSIS

2.1 FUNCTIONAL REQUIREMENTS

- **ADMIN LOGIN:** This feature used by the admin to login into system. They are required to enter a username and password before they are allowed to enter the system. The username and password will be verified and if invalid username is there user is allowed to not enter the system.
- **BOOK ENTRY:** In this module we can store the details of the books .
- **REGISTER STUDENT:** In this module we can keep the details of the new students .
- **REGISTER FACULTY:** In this module we can keep the details of the new faculties .
- **BOOK ISSUE:** This module is used to keep a track of book issue details.
- **BOOK RETURN:** This module enables to keep a track of return books. If the students returns the book after return date fine generates with respective to the days .

2.2 DATA REQUIREMENTS

- The input consists of the query to the database and the outputs consists of the solution for the query .
- The output also includes the user receiving the details of their accounts .
- In this project the inputs will be the queries as fired by the users Like create an accounts, selecting books and putting into accounts.
- Simultaneously inserted values are stored into the database.

2.3 SOFTWARE TOOLS USED

The whole Project is divided into two parts, the front end and the back end.

FRONTEND :

The front end design is using Tkinter GUI in Python3.

- **Python GUI – Tkinter:**

Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter:

- Importing the module – tkinter
- Create the main window (container)
- Add any number of widgets to the main window
- Apply the event Trigger on the widgets.

BACKEND :

The back end is designed using mysql which is used to design the databases.

- **SQLITE3 :**

SQLite is an in-process library that implements a self-contained,serverless,zero-configuration,transaction SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private.SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indexes, triggers, and views, is contained in a single disk file.

CHAPTER - 3

SYSTEM ANALYSIS & DESIGN

System analysis and design deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together. System analysis and design to solve business problems through analyzing the requirements of information systems and designing such systems by applying analysis and design techniques.

System analysis and design is the most essential phase in the development of a system since the logical system design arrived at as a result of systems analysis which is in turn converted into physical system design.

3.1 SYSTEM ARCHITECTURE

Architecture of Library Management System:

- Add or Create Students, Faculties and Books details
- Delete Student and books details
- Returning books
- Search Book category

Basic architecture of library management system is given below:

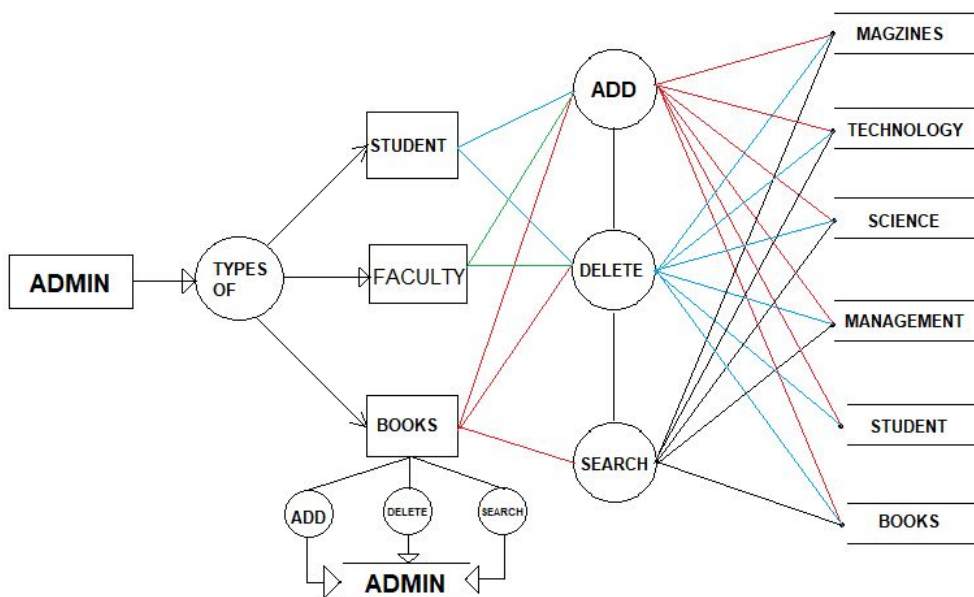


Fig. 3.1: Architecture of library Management System

3.2 SCHEMA DIAGRAM

A schema contains schema objects, which could be tables, columns, data types, views, stored procedures, relationships, primary keys, foreign keys, etc.

A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

SCHEMA DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM:

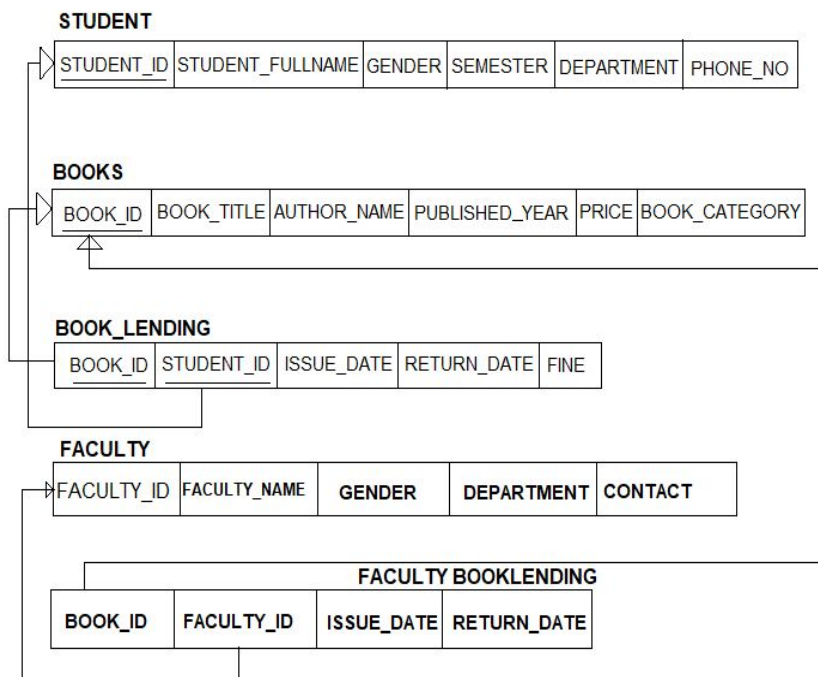


Fig. 3.2: SCHEMA Diagram for Library Management System

3.3 DATA FLOW DIAGRAMS OF LIBRARY MANAGEMENT SYSTEM

The proposed idea for the Library Management System will be represented using data flow diagrams. A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

3.3.1 DFD Level 0 of Initial Data

This is the zero level DFD of library Management System, where we have elaborated the high level process of Library Management .Its a basic overview of the whole library Management System or processed being analysed or modeled . It's designed to be an at-a-glance view of Student showing the system as a single high-level process .

- Managing all the library
- Managing all the book
- Managing all the issue book
- Managing all the student
- Managing all the faculties

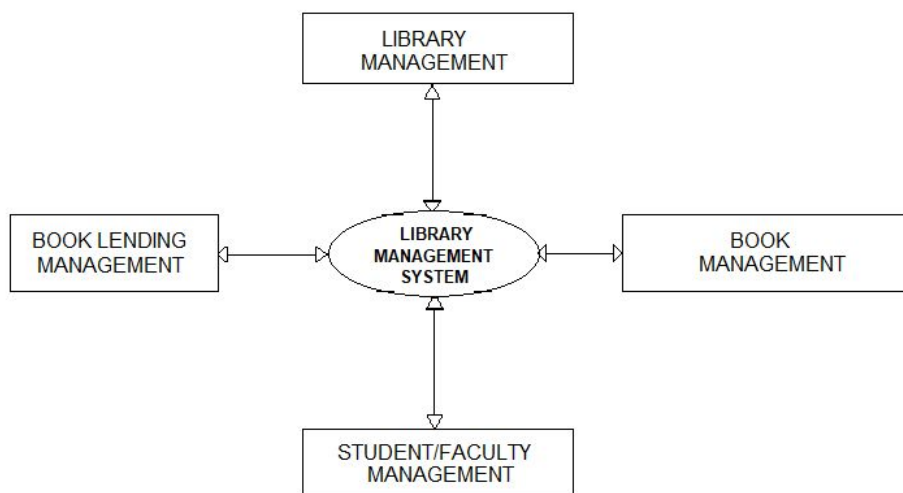


Fig. 3.3.1: DFD Level 0 Of Initial State

3.3.2 DFD Level 1 of Main Page

First Level DFD (1st Level) of Library Management system shows how the system is divided into sub-system (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the library Management System as a whole. It also identifies internal data stores of Student.

Main entities and output of first level DFD (1st Level DFD):

1. Processing Book records and generate a report of all Book.
2. Processing Issue book records and generate a report of all Issue Book.
3. Processing Student records and generate a report of all Student.
3. Processing Faculty records and generate a report of all Faculties.

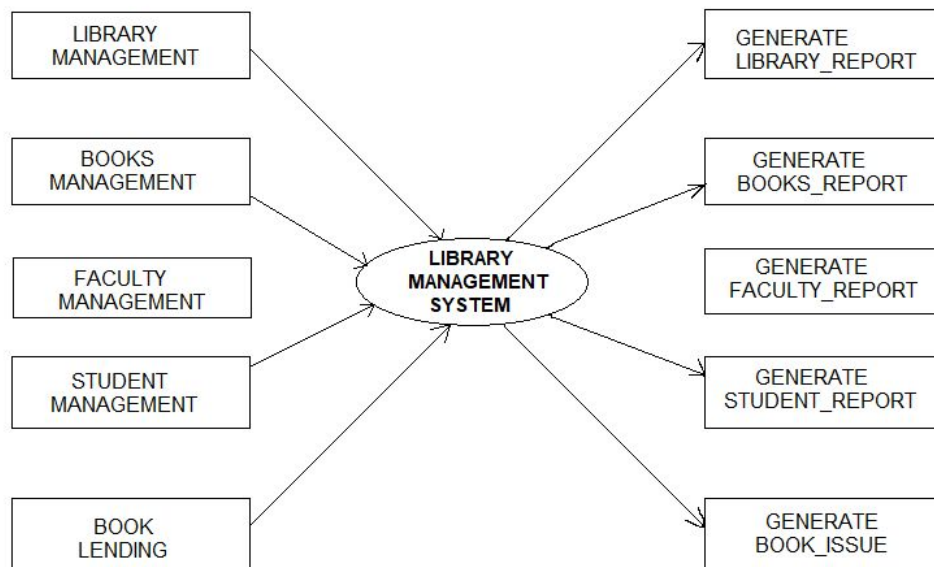


Fig. 3.3.2: DFD Level 1 Of Main Page

3.4 ACTIVITY DIAGRAM

Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation. It is also called object-oriented flowchart. Activity diagrams consist of activities that are made up of actions which apply to behavioral modeling technology.

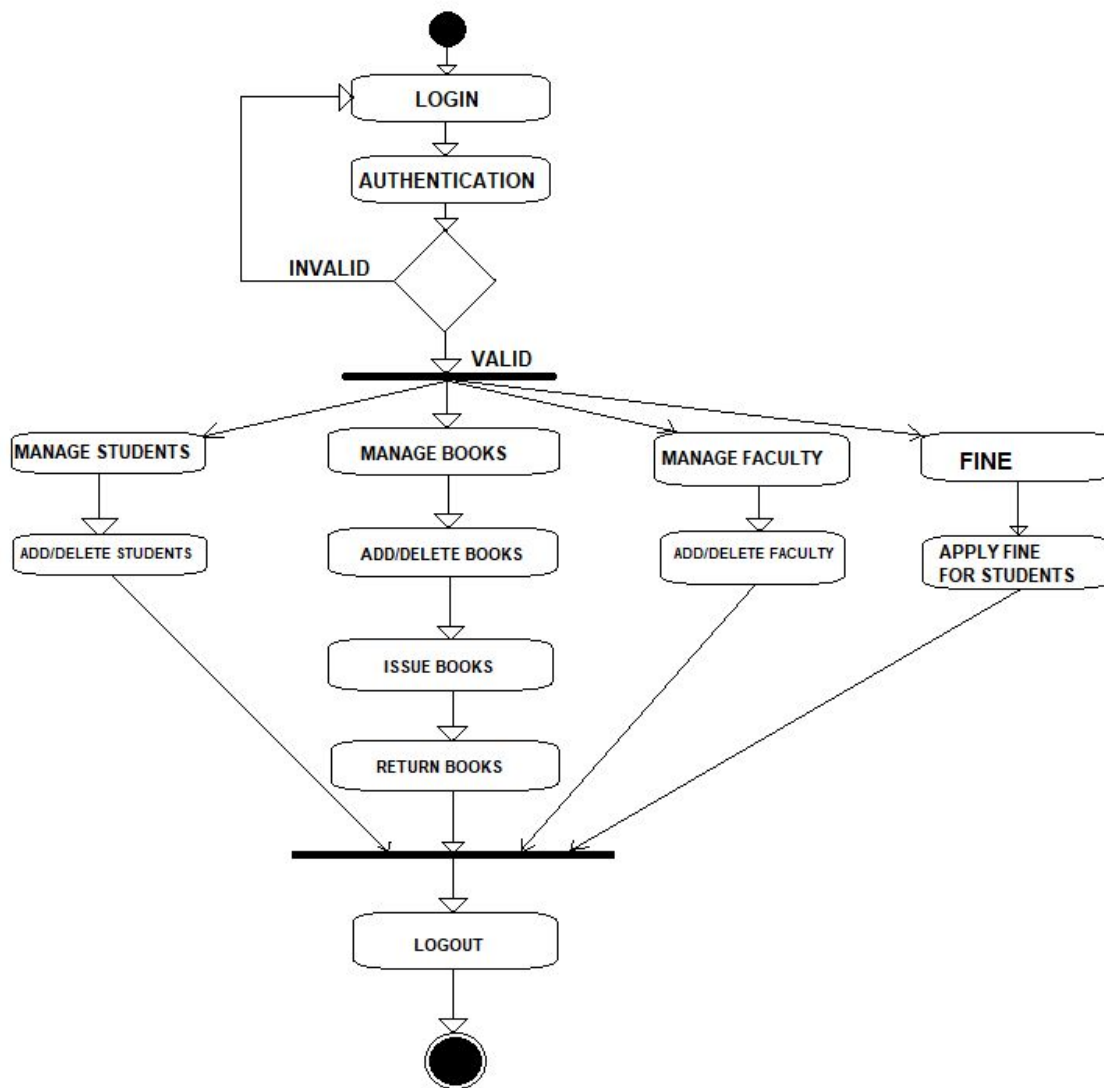


Fig. 3.4: Activity Diagram for Library Management System

3.5 ENTITY-RELATION DIAGRAM

Entity-Relationship Diagram depicts the relationship between the various entities involved. An entity is any real-world object. The attribute of each entity noted in the entity-relationship diagram can be described using an entity description as shown in Fig 3.5

E-R diagram serves two purposes:

- To provide an indication of how data are transformed as they move through the system.
- To depict the functions that transformation the data flow.

Entities

An entity or a data object is the representation of almost any composite information that must be understood by the software. By composite information, we mean something that has a number of different properties or attributes. A data object encapsulates data only. There is no reference within a data object to operation that act on the data.

Attributes

Attributes define the properties of data objects and take on one of three different characteristics. They can be used to:
Name an instance of object. Describe the instance. Make reference to another instance in the table.

Entity-Set and Keys

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.

For example, the roll_number of a student makes him/her identifiable among students.

Super Key – A set of attributes (one or more) that collectively identifies an entity in an entity set.

Candidate Key – A minimal super key is called a candidate key. An entity set may have more than one candidate key

Primary Key – A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

Relationships

Entities are connected to one another and are dependent on one another for certain data transactions. This connection is shown by means of a relationship between the entities involved. The relationship is binary if the number of participating entities are 2, ternary if 3 and m-ary if m.

Examples of Relations:

- 1) Student / Faculty takes Books - M : N
- 2) Books offers Book Lending - M : 1
- 3) Book Lending to Student - 3 : 1
- 4) Book Lending to Faculty - 3 : 1

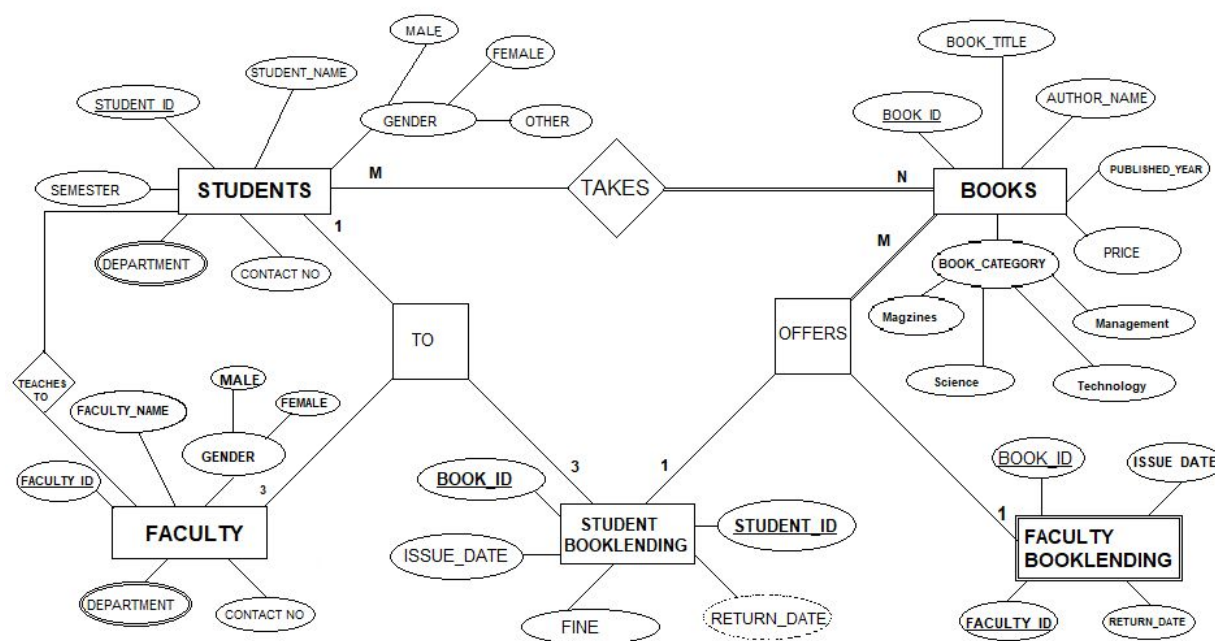


Fig. 3.5: ER Diagram of Library Management System

3.6.1 NORMALIZATION

Database normalization is the process of removing redundant data from your tables. In order to improve storage efficiency, data integrity, and scalability.

Normalization generally involves splitting existing tables into multiple ones, which must be re-joined or linked each time a query is issued.

3.6.1 1ST NORMAL FORM:

Given tables is converted to its 1NF as follows:

- Elimination of duplicative columns from table 1.
- Create separate tables for each group of related data and identify each row with unique column (primary key) .
- create separate table for each group of related data and identify each row with unique column (primary key) .

Table for 1NF:

STUDENT_ID	BOOK_ID	ISSUE_DATE
11	101, 102, 103	21-JAN-2019

AFTER 1NF IN DATABASE:

STUDENT_ID	BOOK_ID	ISSUE_DATE
11	101	21-JAN-2019
11	102	21-JAN-2019
11	103	21-JAN-2019

Fig. 3.6.1: 1st Normal Form

3.6.2 2ND NORMAL FORM:

- A table is in first normal form and each non-key field is functionally dependent upon Primary key
 - It must not contain any partial dependency .
 - Now we'll take the table above and design new tables that will eliminate the repeated Date in no key_field.
- create a table for those fields and enter the sample data.

BOOK_ID	STUD_ID	NAME	BOOK_CATEGORY	ISSUE_DATE	RETURN_DATE
11	101	AAA	FIRST SEM	30-JAN-2019	17-FEB-2019
12	101	AAA	SECOND SEM	21-FEB-2019	28-FEB-2019
13	102	BBB	THIRD SEM	12-JUN-2019	20-JUN-2019

AFTER 2ND NORMAL FORM:

BOOK_ID	STUD_ID	NAME	BOOK_CATEGORY
11	101	AAA	FIRST SEM
12	101	AAA	SECOND SEM
13	102	BBB	THIRD SEM

BOOK_ID	BOOK_CATEGORY	ISSUE_DATE
11	FIRST SEM	30-JAN-2019
12	SECOND SEM	21-FEB-2019
13	THIRD SEM	12-JUN-2019

STUD_ID	ISSUE_DATE	RETURN_DATE
101	30-JAN-2019	17-FEB-2019
101	21-FEB-2019	28-FEB-2019
102	12-JUN-2019	20-JUN-2019

Fig. 3.6.2: 2nd Normal Form

3.6.3 3RD NORMAL FORM (3NF):

Third normal form (3NF) requires that there are no functional dependency of non-key. Attributes on something other than a candidate key .

- A table is in 3NF if all of the non primary-key attributes are mutually independent .
- There should not be transitive dependencies .

➤ create a table for those fields and enter the sample data.

STUD_ID	BOOK_ID	NAME	GENDER	BOOK_CATEGORY
101	11	AAA	MALE	FIRST SEM
101	12	AAA	MALE	SECOND SEM
102	13	BBB	FEMALE	THIRD SEM

AFTER 3NF :

STUD_ID	NAME	GENDER
101	AAA	MALE
101	AAA	MALE
102	BBB	FEMALE

BOOK_ID	NAME	BOOK_CATEGORY
11	AAA	FIRST SEM
12	AAA	SECOND SEM
13	BBB	THIRD SEM

Fig. 3.6.3: 3rd Normal Form

CHAPTER - 4

RESULTS

A Single user, usually the admin could successfully log-in by entering his username and password which will lead to the main page or else if entered wrong password or username it shows “invalid username or password”.

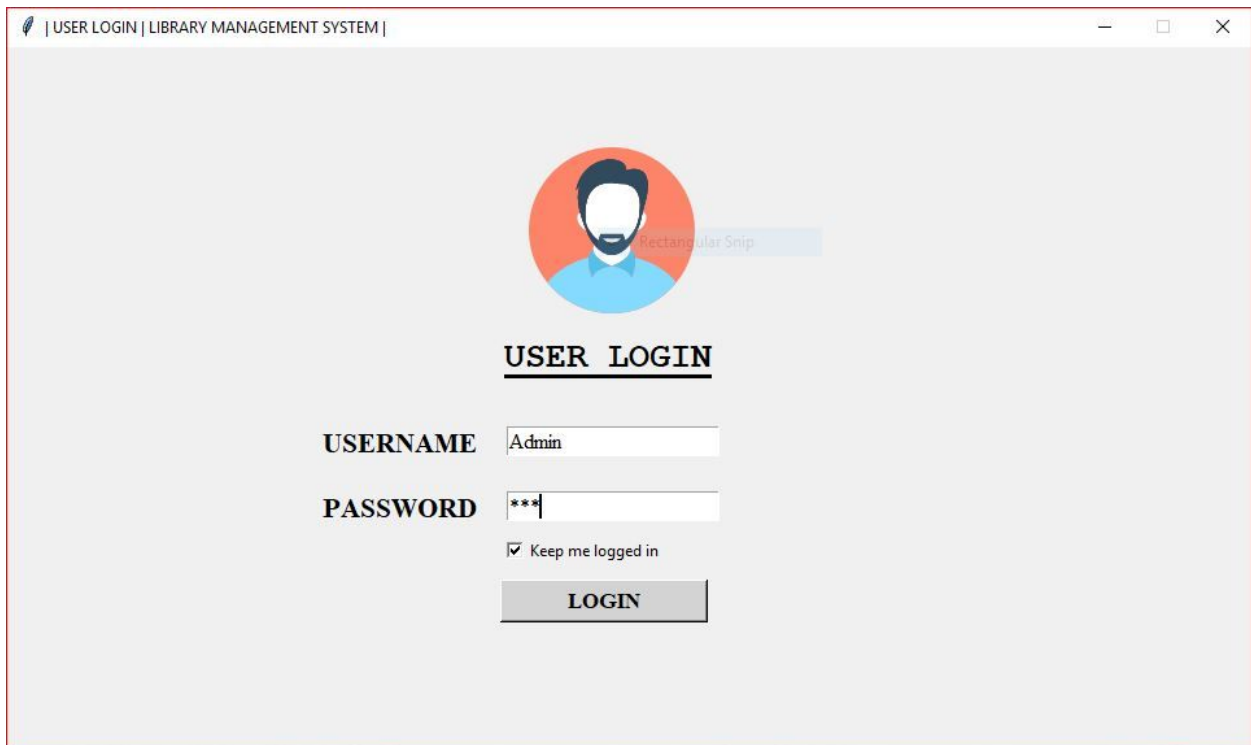
The main page provides links to different data such as students, books and book lending data where the admin could add any new student data by providing particulars of a students such as id, name, sem, department, gender, phone no., and press “submit”. The student update option could successfully search for the student details that needs to be updated by taking the unique id of student and the details of the student of given id no could be updated as desired. The student data of a particular student was deleted by giving the id of student. All these details of student is showed up on student data table.

The books table has an option as “add/delete”, which adds new books information such as book_id, title, author_name, published_year, price and book_category given by the admin, and press “submit”. These details showed up in the books data table.

The book_lending table has options to add, delete and view the book_lending data that is entered. In the “book_lending” table the admin could enter the book_id, student_id and press “submit”. After submitting the data, the table contains issue_date, return_date, and fine it will take current date of the issue_date . The books data could be deleted in the delete option by giving the unique id of the book. These details showed up in the books data table.

After the actions the user could successfully log-out of the system. Thus, the above features were tested and is free from defects.

4.1 SNAPSHOTS



The screenshot shows a web browser window titled "USER LOGIN | LIBRARY MANAGEMENT SYSTEM |". The page has a light gray background. At the top center is a circular profile picture of a man with a beard, wearing a blue shirt. Below the picture is the text "Rectangular Snip". Underneath the picture is the heading "USER LOGIN" in bold, underlined black text. Below the heading are two input fields: "USERNAME" with the text "Admin" and "PASSWORD" with three asterisks "***". Below the password field is a checkbox labeled "Keep me logged in" which is checked. At the bottom is a gray button with the text "LOGIN" in bold black text.

Fig. 7.1.1: Login Page

Login page allows user to enter username and password for verification purposes to keep the software safe from unauthorised access.

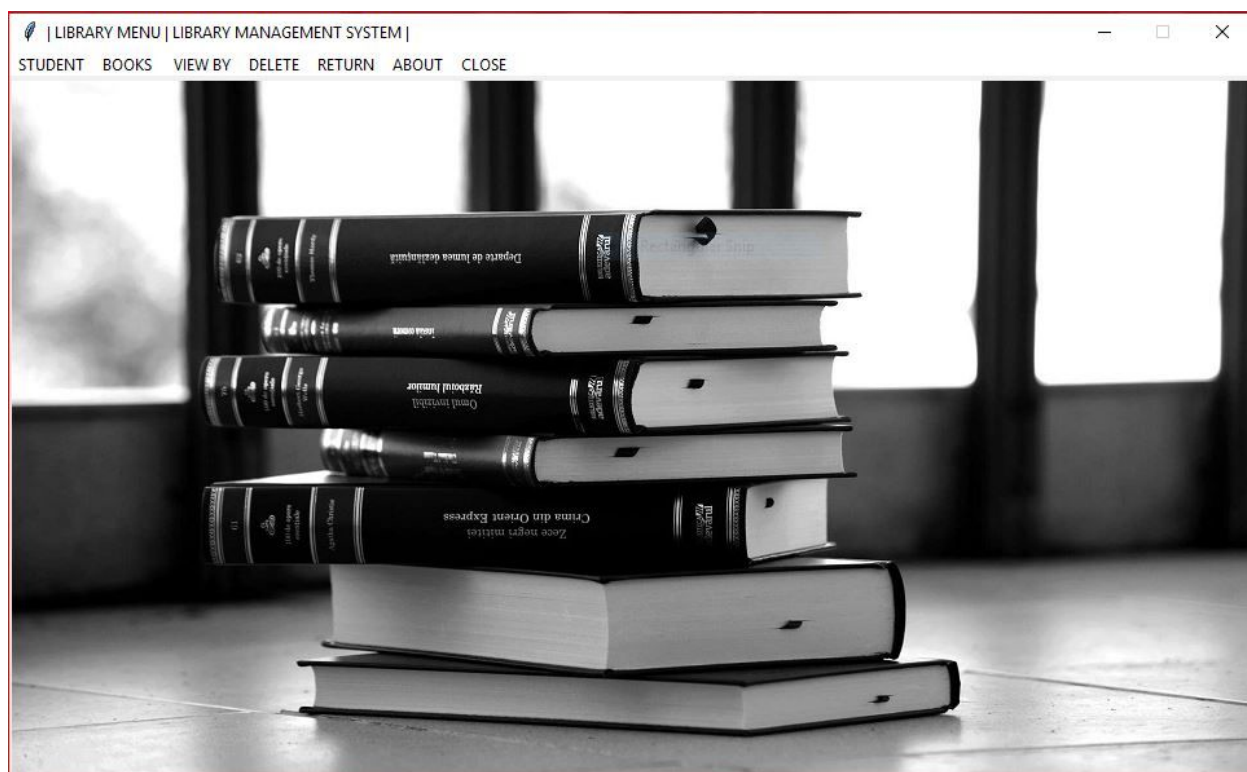


Fig. 7.1.2: Main Page

Main page allows access to different data and provides link to add new details to the tables

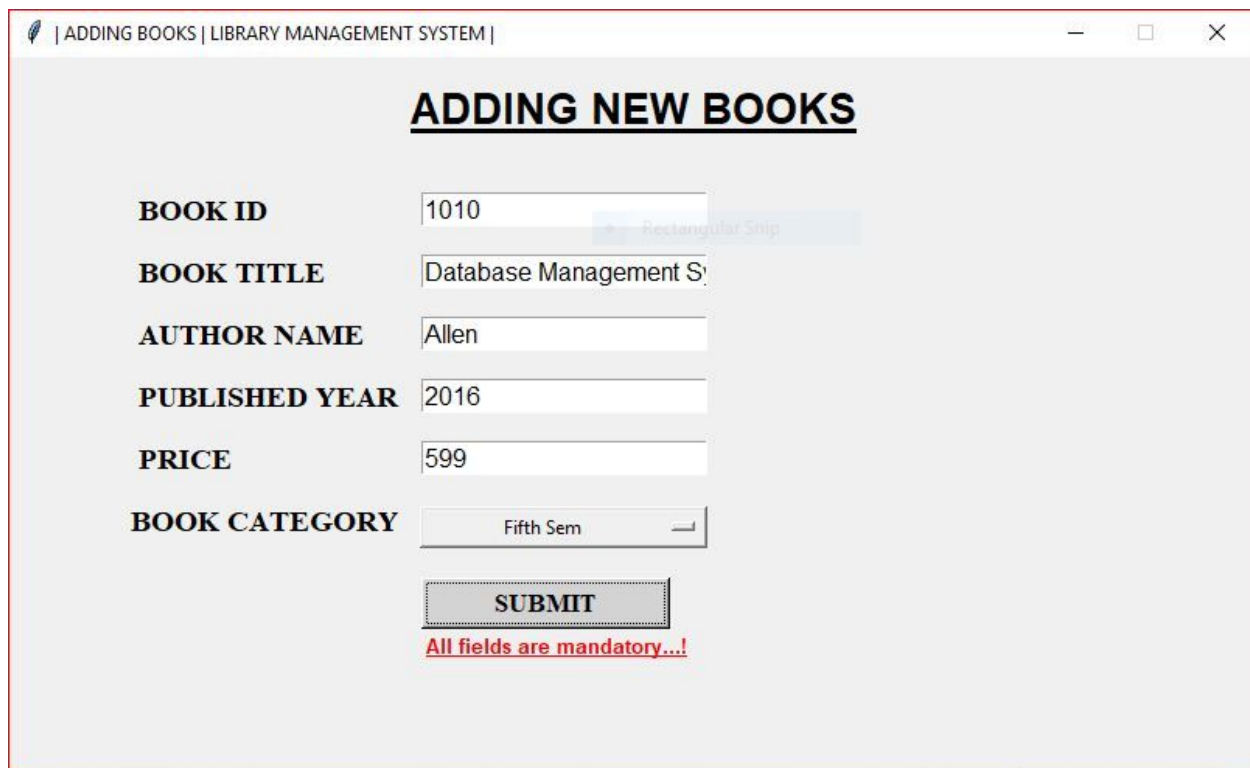
STUDENT REGISTRATION

STUDENT ID	<input type="text" value="201"/>
STUDENT NAME	<input type="text" value="Nikith Kumar N"/>
GENDER	<input type="text" value="Male"/>
SEMESTER	<input type="text" value="Fifth"/>
DEPARTMENT	<input type="text" value="CSE"/>
CONTACT NO.	<input type="text" value="9632587410"/>

SUBMIT

All fields are mandatory...!

Fig. 7.1.3: Student Information Insertion



| ADDING BOOKS | LIBRARY MANAGEMENT SYSTEM |

ADDING NEW BOOKS

BOOK ID

BOOK TITLE

AUTHOR NAME

PUBLISHED YEAR

PRICE

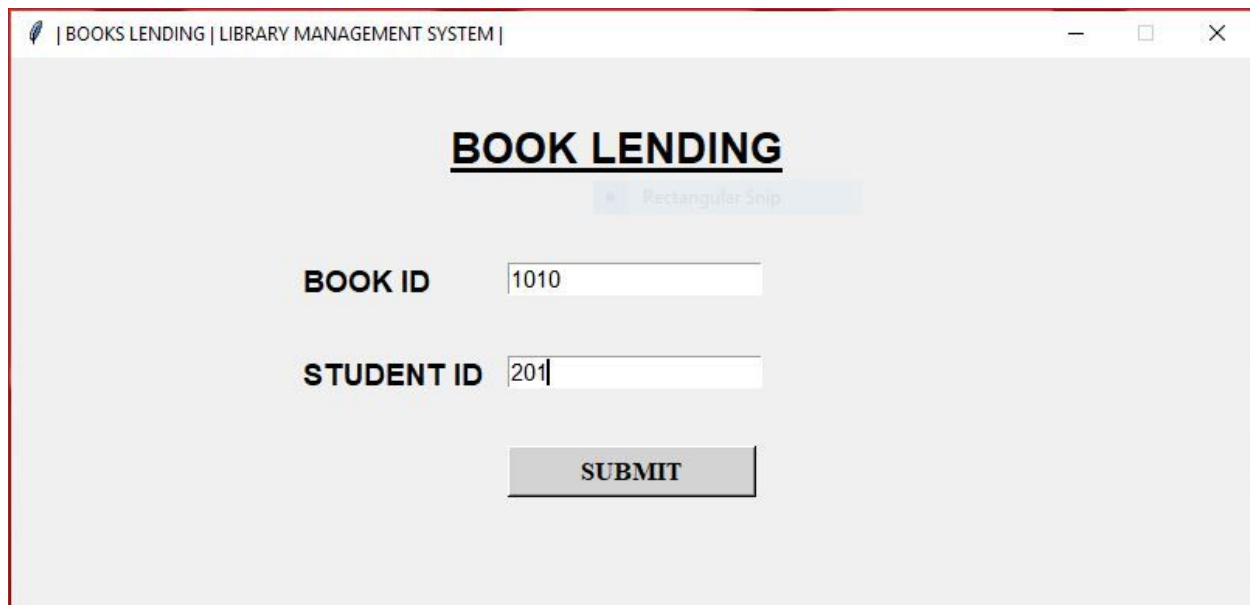
BOOK CATEGORY

SUBMIT

All fields are mandatory...!

Fig. 7.1.4: Book Details

Book details shows details that have to be added to book table



| BOOKS LENDING | LIBRARY MANAGEMENT SYSTEM |

BOOK LENDING

BOOK ID

STUDENT ID

SUBMIT

Fig. 7.1.5: Book Lending Details

Book Lending details shows details that have to be added to book lending table

| VIEWING STUDENT DETAILS | LIBRARY MANAGEMENT SYSTEM |

VIEW STUDENT DETAILS

STUDENT_ID	FULL_NAME	GENDER	SEMESTER	DEPARTMENT	PHONE_NO
101	Jack	Male	Second	ISE	9632587410
103	Jill	Male	Third	CSE	7521046983
104	Natasha	Female	Second	ME	965234017
105	Black	Male	Third	CIV	6210648975
106	Thor	Male	Fifth	ME	8520314679
107	Bruce	Male	Fifth	CSE	7337637319
108	Tony	Male	Fifth	CSE	9662458775
109	Captian	Male	Third	CIV	7316485206
110	Doctor	Female	Second	ME	7411528963
201	Nikith Kumar N	Male	Fifth	CSE	9632587410

Fig. 7.1.6: Student Details

Student details page shows the data that must be entered about the student in the library

| VIEWING BOOK DETAILS | LIBRARY MANAGEMENT SYSTEM |

VIEW BOOK DETAILS

BOOK_ID	BOOK_TITLE	AUTHOR NAME	PUBLISHED_YEAR	PRICE	BOOK_CATEGORY
100	AAA	aaa	2001	299	QWERT
101	BBB	bbb	2002	399	ASDFG
102	CCC	ccc	2003	499	Fourth
103	DDD	ddd	2004	449	Fourth
104	EEE	eee	2005	559	Fourth
106	FFF	fff	2007	669	Fourth
108	GGG	ggg	2008	569	First
109	HHH	hhh	2012	696	Second
1010	Database Manageme	Allen	2016	599	Fifth Sem

Fig.7.1.7: Books Details

Books details page shows the data that must be entered about the books in the library

BOOK LENDING DETAILS LIBRARY MANAGEMENT SYSTEM			
<u>VIEW BOOK LENDING DETAILS</u>			
BOOK_ID	STUDENT_ID	ISSUE_DATE	RETURN_DATE
203	104	11/04/2019	11/11/2019
106	205	11/04/2019	11/11/2019
108	206	11/04/2019	11/11/2019
1010	201	11/10/2019	11/17/2019

Fig. 7.1.8: Book lending Details

Book Lending details page shows the data that must be entered about the book lending in the library

DELETING STUDENT DETAILS LIBRARY MANAGEMENT SYSTEM						
<u>DELETING STUDENT DETAILS</u>						
STUDENT_ID	FULL_NAME	GENDER	SEM	DEPARTMENT	PHONE NO	DELETE
101	Jack	Male	Second	ISE	9632587410	DELETE
103	Jill	Male	Third	CSE	7521046983	DELETE
104	Natasha	Female	Second	ME	965234017	DELETE
105	Black	Male	Third	CIV	6210648975	DELETE
106	Thor	Male	Fifth	ME	8520314679	DELETE
107	Bruce	Male	Fifth	CSE	7337637319	DELETE
108	Tony	Male	Fifth	CSE	9662458775	DELETE
109	Captian	Male	Third	CIV	7316485206	DELETE
110	Doctor	Female	Second	ME	7411528963	DELETE
201	Nikith Kumar N	Male	Fifth	CSE	9632587410	DELETE

Fig. 7.1.9: Student Details Deletion

Student details deletion page shows the data that must be deleted in the student database.

CONCLUSION AND FUTURE ENHANCEMENT

The system has been developed with much care that it is free of errors and at the same time it is efficient and less time consuming. The important thing is that the system is robust. It avoids malfunction from outsiders. It goes through all phases of software development cycle. So, product is accurate. Also, provision is provided for future developments in the system.

The Library Management system can be used for the proper functioning of the sanctuary. It provides Information to the User about the library system, through which he/she can borrow books from the library. The library system allows student to borrow only 3 books at once, if he/she takes more than 3 books the system show error message ie; (This student_id has taken 3 books please return back and borrow another on). This system calculates fine for book lending after the date exceeds ie; 7days from issue date, if return date is expired it calculates per day 2Rs automatically. After all the operation is over the admin can logout from the system.

Further enhancements to the websites can be made, making it more user friendly, more innovation can be brought out, instead of sticking into fixed boundaries. This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when required in this application. There is flexibility in all the modules

REFERENCES

Websites:

- [1] W3 Schools, Tkinter Tutorials: <http://www.w3schools.com>
- [2] Google Chrome
- [3] Youtube
- [4] <http://www.stackoverflow.com>
- [5] url = '<https://docs.python.org/3.3/library/tk.html>'
- [6] url = '<https://pythonprogramming.net/python-3-tkinter-basics-tutorial>'
- [7] Tkinter videos: <https://youtu.be/e3i5tMxmcxk>

APPENDIX

SOURCE CODE

- Code for frontend Admin Login Page

This is the login page of Library Management System..

```
from tkinter import *
import Third Page
from PIL import ImageTk, Image
import tkinter.messagebox as nk

class MainWindow:

    def __init__(self):
        self.win = Tk()
        # window background color using canvas
        self.canvas = Canvas(self.win, width=960, height=540, bg='white')
        self.canvas.pack(expand=YES, fill=BOTH)

        # show window in the center of the screen
        width = self.win.winfo_screenwidth()
        height = self.win.winfo_screenheight()
        x = int(width / 2 - 960 / 2)
        y = int(height / 2 - 540 / 2)
        str1 = "960x540+" + str(x) + "+" + str(y)
        self.win.geometry(str1)

        # disable resize window
        self.win.resizable(False, False)

        # changing the title of the window
        self.win.title("| USER LOGIN | LIBRARY MANAGEMENT SYSTEM |")
```

```
def add_frame(self):
    self.frame = Frame(self.win, height=540, width=960)
    self.frame.place(x=0, y=0)

self.image =
ImageTk.PhotoImage(Image.open("C:\\Users\\Nikith\\PycharmProjects\\Bolt2.py\\images
\\man.png"))
    self.label = Label(self.frame, image=self.image)
    self.label.place(x=400, y=75)

    # creating login form for only admin can access
    self.label = Label(self.frame, text="USER LOGIN")
    self.label.config(font=("Courier", 20, 'underline bold'))
    self.label.place(x=380, y=220)

    # creating username label and entry field
    self.label1 = Label(self.frame, text="USERNAME")
    self.label1.config(font=("Times", 16, 'bold'))
    self.label1.place(x=240, y=290)

    self.user = Entry(self.frame, font="Times 12")
    self.user.place(x=385, y=292)

    # creating password label and entry field
    self.label2 = Label(self.frame, text="PASSWORD")
    self.label2.config(font=("Times", 16, 'bold'))
    self.label2.place(x=240, y=340)

    self.pswd = Entry(self.frame, font="Times 12", show="*")
    self.pswd.place(x=385, y=342)

    self.checkbox = Checkbutton(self.frame, text="Keep me logged in").place(x=380,
y=375)

    self.btn = Button(self.frame, text='LOGIN', width=15, bg='light grey', fg='black',
font=("Times", 13, "bold"),
        command=self.login
```



```
self.btn.place(x=380, y=410)
def login(self):
    data = (
        self.user.get(),
        self.pswd.get()
    )

    # if else condition for user authenticate..!
    if self.user.get() == "Admin" and self.pswd.get() == "160698":
        nk.showinfo("Login info", "Welcome To \n Library Management System...!")
        self.win.destroy()
        t = Thirdpage.ThirdWindow()
        t.add_menu()
        t.add_frame()
    else:
        nk.showinfo("Login error", "Invalid Username & Password")

    self.win.mainloop()
```

- Code for backend database

```
import sqlite3
import datetime

# Creating and inserting student details..!
def StudentReg(data):
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute(
        'CREATE TABLE IF NOT EXISTS Student ("STUDENT_ID" INTEGER UNIQUE,
        "FULL_NAME" TEXT, "GENDER" TEXT, '
        '"SEM" TEXT, "DEPARTMENT" TEXT, "PHONE NO" INTEGER, PRIMARY KEY("STUDENT
        ID"))')
    cur.execute('INSERT INTO Student VALUES(?,?,?,?,?,?)', data)
    conn.commit()

# Creating and inserting book details..!

def AddBook(data):
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute(
        'CREATE TABLE IF NOT EXISTS Books ("BOOK_ID" INTEGER, "BOOK_TITLE" TEXT,
        "AUTHOR_NAME" TEXT, '
        '"PUBLISHED_YEAR" INTEGER, "PRICE"    INTEGER, PRIMARY KEY("BOOK_ID"))')
    cur.execute('INSERT INTO Books VALUES(?,?,?,?,?)', data)
    conn.commit()

# Creating and inserting book details..!

def BookLending(data, id):
    status = 0
    id_status = 0
    print(data, type(data[3]))
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
```

```
cur.execute(
    'CREATE TABLE IF NOT EXISTS BookLending ("BOOK_ID" references Books on DELETE
    CASCADE, "STUDENT_ID"
        'references Student on DELETE CASCADE, "ISSUE_DATE" INTEGER, "RETURN_DATE"
    INTEGER, "FINE" INTEGER, '
        'PRIMARY KEY("BOOK_ID", "STUDENT_ID"))')
cur.execute('SELECT COUNT(*) FROM BookLending B, Student S where B.STUDENT_ID =
? and S.STUDENT_ID = ?', id)
id_count = cur.fetchone()
print(type(id_count), id_count[0])
if id_count[0] <= 2:
    id_status = 1
    cur.execute('INSERT INTO BookLending VALUES(?,?,?,0)', data)
else:
    id_status = 0
status = 1
conn.commit()
return int(status), int(id_status)

def ViewStudents():
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute("SELECT * FROM Student")
    return cur.fetchall()

def ViewBooks():
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute("SELECT * FROM Books")
    return cur.fetchall()

def BookLend():
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute("SELECT * FROM BookLending")
    return cur.fetchall()
```

```
def DeleteStudent(data):
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute('DELETE FROM Student WHERE STUDENT_ID = ?', data)
    conn.commit()
    return True

def Update(data):
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute('UPDATE Books SET BOOK_ID = ?, BOOK_TITLE = ?, AUTHOR_NAME = ?,
PUBLISHED_YEAR = ?, PRICE = ? WHERE '
                'BOOK_ID = ?', data)
    conn.commit()
    return True

def Delete(data):
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    cur.execute('DELETE FROM Books WHERE BOOK_ID = ?', data)
    conn.commit()
    return True

def Return(data):
    conn = sqlite3.connect("Library.db")
    cur = conn.cursor()
    print(data)
    date_format = "%m/%d/%Y"
    today = datetime.date.today()
    current_date = str(today.strftime("%m/%d/%Y"))
    print(type(current_date))
    cur.execute('select return_date from BookLending where book_id = ? and student_id =
?', data)
    return_date = cur.fetchone()
    fine_days = (datetime.datetime.strptime(current_date, date_format) -
datetime.datetime.strptime(return_date[0], date_format)).total_seconds() / 60 / 60 / 24
    fine_days = int(fine_days)
```

```
cur.execute("delete from BookLending where book_id=? and student_id=?", data)
print(fine_days)
if fine_days <= 0:
    fine_days = 0
conn.commit()
return True, fine_days
```