

**Engineering Exploration Project report on**

**LIBRARY MANAGEMENT SYSTEM**

# **GOPAL MATCHA 1601-20-733-032**

**SAI KISHORE BANDARI 1601-20-733-044**

**TANISH ROHIL GALI 1601-20-733-056**

# 

Under the mentorship of **MS. TALLA PRASHANTHI**, Asst. Prof.

# **Core branch: CSE (CSE – 1)**

**Submitted to**

**Computer Science Engineering Department**

# **Chaitanya Bharathi Institute of Technology (A)**

**ABSTRACT:**

The Library Management System is an application for assisting a librarian in managing a book library in a university. The system would provide basic set of features to add/update members, add/update books, and manage check in specifications for the systems based on the client’s statement of need.

The project titled Library Management System is for monitoring and controlling the transactions in a library. The project “Library Management System” is developed in php, which mainly focuses on basic operations in a library like adding new books, updating new information, searching books, searching members and returning books.

This project of “LIBRARY MANAGEMENT” gives us the complete information about the library. We can enter the record of new books and retrieve the details of books available in the library. We can issue the books to the students and maintain their records and can also check how many books are issued and stock available in the library. In this project we can maintain the late fine of students who returns the issued books after the due date.

.

**CONTENTS**

**PAGE NO:**

ABSTRACT 01

LIST OF FIGURES. 03

1. INTRODUCTION TO LIBRARY MANAGEMENT SYSTEM: 04
   1. Introduction
   2. Objective of the project
   3. Modules

2. THEORETICAL ANALYSIS OF LIBRARY MANAGEMENT SYSTEM 05

2.1 Existing system

2.2 Proposed system

2.3 Feasibility Analysis

2.3.1 Technical feasibility

2.3.2. Economic feasibility

3. ANALYSIS AND OF LIBRARY MANAGEMENT SYSTEM. 08

3.1 System requirements

3.1.1 Hardware configuration

3.1.2. Software configuration

3.2 Data flow diagram

4. IMPLEMENTATION. 11

4.1 Source code

4.2 Description of modules

5. RESULT AND DISCUSSION 16

5.1 Screenshots

6. CONCLUSION AND FUTURE ENHANCEMENT. 18

6.1 Conclusion

6.2 Future Enhancement

REFERENCES. 19

**LIST OF FIGURES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO.** | **FIGURE NO** | **FIGURE NAME** |  | **PAGE NO.** | |
|  |  |  |  |  | |
| 1 | 2.1 | Architecture of Proposed System |  | 06 | |
| 2 | 3.1 | Data Flow Diagram |  | 09 | |
| 3 | 3.2 | Level-1: Various tasks that can be done using this system. | | | 09 |
| 4 | 3.3 | Level-2: Various ways in which the system can be used by the user | | | 10 |
| 5 | 3.4 | Level-3: Various ways in which the system can be configured by the admin. | | | 10 |

1.INTRODUCTION OF LIBRARYMANAGEMENT SYSTEM

**1.1 INTRODUCTION:**

The project titled Library Management System is Library management software for monitoring and controlling the transactions in a library[8]. The project “Library Management System” is developed in python, which mainly focuses on basic operations in a library like adding new member, new books, and updating new information, searching books and members and facility to borrow and return books[6]. “Library Management System” is a windows application written for 32-bit Windows operating systems, designed to help users maintain and organize [3]. Our software is easy to use for both beginners and advanced users[7]. It features a familiar and well thought-out, an attractive user interface[9]. This system excludes the use of paper work by managing all the book information electronically[2]. Admin can keep updating the system by providing the new books arrival in system and their availability thus students need not to go to library for issuing purpose[1]. The system has books which are well organized and systematically arranged in different categories in the system so that user can easily search and find the book. Thus, it saves human efforts and resources[4].Library management system is a typical management Information system (MIS), its Development include the establishment and maintenance of back-end database and front-end application development aspects[5].

**1.2 OBJECTIVE OF THE PROJECT:**

* More Accurate
* Speed improvement
* Data inconsistency
* Better error handling
* Integrity
* The System will handle the maintenance of data.
* It handles entire transaction of book registration till issue of the book.
* Administrator will immediately acknowledge with error message.
* The Administrator details of Books, Authors and existing Member etc. Book wise search can be performed.

**1.3 MODULES:**

The software Library Management System has two main modules:

1. Insertion to Database Module – User friendly input screen

2. Search Facility system – search for books

2. **THEORITICAL ANALYSIS OF LIBRARY MANAGEMENT SYSTEM**:

**2.1 EXISTING SYSTEM:**

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system [5]. Here the key question is- what all problems exist in the present system? What must be done to solve the problem?[6]. Analysis begins when a user or manager begins a study of the program using existing system[2].During analysis, data collected on the various files, decision points and transactions handled by the present system[3]. The commonly used tools in the system are Data Flow Diagram, interviews, etc. [4]. Training, experience and common sense are required for collection of relevant information needed to develop the system [7].The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution [8].A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution [1].Thus, it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs[9].

**System analysis can be categorized into four parts:**

1. System planning and initial investigation
2. Information Gathering
3. Applying analysis tools for structured analysis
4. Feasibility study ¸ Cost/ Benefit analysis.

In our existing system all the transaction of books are done manually.So, taking more time for a transaction like borrowing a book or returning a book and also for searching of members and books. Another major disadvantage is that to preparing the list of books borrowed and the available books in the library will take more time, currently it is doing as a one-day process for verifying all records. So, after conducting the feasibility study we decided to make the manual Library management system to be computerized.

**DISADVANTAGES:**

1. Since, all the transactions of books are done manually, a lot of time is taken
2. Searches Take Longer and they are Less Efficient
3. Vulnerability to Human Error
4. Costs Add Up Over Time

**2.2 PROPOSED SYSTEM**:

Proposed system is an automated Library Management System. Through our software user can add members, add books, search members, search books, update information, edit information, borrow and return books in quick time.

**2.2.1 ARCHITECTURE OF PROPOSED SYSTEM:**

Return a Book

Borrow a Book

Search for a Book

Check the rent and fine

View the existing books

Library Management

System

Request

Software

Response

USER PROGRAM

**Fig 2.1: Architecture of Proposed System**

**2.2.2 ADVANTAGES:**

Our proposed system has the following advantages:

1. User friendly interface
2. More Storage Capacity
3. Search facility
4. Quick transaction
5. All the manual difficulties in managing the library have been rectified by implementing computerization

**2.3 FEASIBILITY ANALYSIS:**

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible.

Here the feasibility study can be performed in two ways such as:

Technical feasibility and Economical Feasibility.

* + 1. **TECHNICAL FEASIBILITY:**

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

**2.3.2 ECONOMICAL FEASIBILITY:**

Development of this application is highly economically feasible. The organization need not spend much more for the development of the system already available. The only things to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even after the development, the organization will not be in a condition to invest more in the organization. Therefore, the system is economically feasible.

3. **ANALYSIS AND DESIGN OF LIBRARY MANAGEMENT SYSTEM**

**3.1.1 HARDWARE CONFIGURATION:**

## Processor : Intel Core i7-5960X or Above

* RAM : 128 MB
* Hard Disk : 20GB   
  Monitor : Color monitor
* Key Board : 122 Keys
* Mouse : (Optional) Any Optical Mouse

**3.1.2 SOFTWARE CONFIGURATION**:

* Language : Python
* Software Platform : Python 3.8(64-bit) or above
* Database : MySQL

Setup tools and pip to be installed for 3.6.x and above

**3.2 DATA FLOW DIAGRAM:**

**Fig 3.1: Data Flow Diagram**

**LEVEL-1:**

**BOOK**

**Fig 3.2: Various tasks that can be done using this system.**

**LEVEL-2:**

**USER**

**Fig 3.3: Various ways in which the system can be used by the user**

**LEVEL-3:**

**ADMIN**

**4**

**Fig 3.4: various ways in which the system can be configured by the admin**

4.**IMPLEMENTATION**:

**4.1 SOURCE CODE:**

1. **// Using Database//**

import sqlite3

conn = sqlite3.connect("database.db")

c = conn.cursor()

c.execute('CREATE TABLE IF NOT EXISTS books (title TEXT, quantity INTEGER, author TEXT, pages INTEGER)')

conn.commit()

def addNewBooks(title, quantity, author, pages):

    c.execute('INSERT INTO books VALUES (?, ?, ?, ?)', (title, quantity, author, pages))

    conn.commit()

def addExistingBooks(title, quantity):

    c.execute('UPDATE books SET quantity = (quantity + ?) WHERE title = (?)', (quantity, title))

    conn.commit()

def removeBooks(title, quantity):

    c.execute('UPDATE books SET quantity = (quantity - ?) WHERE title = (?)', (quantity, title))

    conn.commit()

1. **// LIBRARY MANAGEMENT SYSTEM//**

from tkinter import \*

from datetime import date

# database

from libDatabase import \*

# gui start

root = Tk()

root.title("Library Management")

title = Label(root, text="LIBRARY MANAGEMENT SERVICE")

title.grid(row=0, column=0)

# base window

# top window

def statusCheck():

    check = Toplevel()

    check.title("Currently available")

    title = Label(check, text="Currently available books/journals")

    title.pack()

    c.execute('SELECT \* FROM books')

    for i, j, k, l in c.fetchall():

        status = Label(check, text="There are " + str(j) + " copies of " + "'" + i + "' by " + "'" + k + "' available, it is '" + str(l) + "' pages long.")

        status.pack()

checkStatus = Label(root, text="Do you want to check the available books/journals?", anchor=E)

checkStatus.grid(row=1, column=0)

checkButton = Button(root, text="Show", padx=3, pady=3, command=statusCheck)

checkButton.grid(row=2, column=0)

frame = LabelFrame(root, padx=6, pady=6)

frame.grid(row=3, column=0, padx=6, pady=6)

nameLabel = Label(frame, text="Name: ", anchor=E)

nameLabel.grid(row=0, column=0)

nameEntry = Entry(frame, width=30, borderwidth=3)

nameEntry.grid(row=0, column=1)

authLabel = Label(frame, text="Author: ", anchor=E)

authLabel.grid(row=3, column=0)

authEntry = Entry(frame, width=30, borderwidth=3)

authEntry.grid(row=3, column=1)

pageLabel = Label(frame, text="No. of pages: ", anchor=E)

pageLabel.grid(row=4, column=0)

pageEntry = Entry(frame, width=30, borderwidth=3)

pageEntry.grid(row=4, column=1)

quantityLabel = Label(frame, text="Quantity (in numerals): ", anchor=E)

quantityLabel.grid(row=1, column=0)

quantityEntry = Entry(frame, width=30, borderwidth=3)

quantityEntry.grid(row=1, column=1)

Label(frame, text="What do you want to do?").grid(row=5, column=1)

status = IntVar()

add = Radiobutton(frame, text="Return this book(s) to the library", variable=status, value=1, anchor=E)

sub = Radiobutton(frame, text="Borrow this book(s) from the library", variable=status, value=0, anchor=E)

add.grid(row=6, column=1)

sub.grid(row=7, column=1)

status.set(None)

def confirmation():

    if status.get() == 1:

            c.execute('SELECT title FROM books')

            for i in c.fetchall():

                if i[0] == nameEntry.get():

                    addExistingBooks(nameEntry.get(), int(quantityEntry.get()))

                    Label(root, text=quantityEntry.get() + " copies of " + nameEntry.get() + " successully returned!").grid(row=4, column=0)

                    break

                else:

                    addNewBooks(nameEntry.get(), int(quantityEntry.get()), authEntry.get(), pageEntry.get())

                    Label(root, text=quantityEntry.get() + " copies of " + nameEntry.get() + " successully returned!").grid(row=4, column=0)

    else:

        c.execute('SELECT title, quantity FROM books')

        for i in c.fetchall():

            if i[0] == nameEntry.get():

                if i[1] > 0:

                    if i[1] - int(quantityEntry.get()) >= 0:

                        removeBooks(nameEntry.get(), int(quantityEntry.get()))

                        Label(root, text=quantityEntry.get() + " copies of " + nameEntry.get() + " successully taken!").grid(row=4, column=0)

                        break

                    else:

                        Label(root, text="Borrowing non-existing books not possible!").grid(row=4, column=0)

            else:

                Label(root, text="The book does not exist!").grid(row=4, column=0)

confirm = Button(frame, text="Confirm", padx=3, pady=3, command=confirmation)

confirm.grid(row=8, column=1)

Label(root, text="".center(69, '-')).grid(row=5, column=0)

frame2 = LabelFrame(root, text="Fine checker".center(14), padx=6, pady=6)

frame2.grid(row=6, column=0, padx=6, pady=6)

dateLabel = Label(frame2, text="Date of issue: (DD MM YYYY): ", anchor=E)

dateLabel.grid(row=0, column=0)

dateEntry = Entry(frame2, width=30, borderwidth=3)

dateEntry.grid(row=0, column=1)

def fineCheck():

    day = dateEntry.get().split()

    delta = date.today() - date(int(day[2]), int(day[1]), int(day[0]))

    Label(frame2, text="Your fine is " + str((delta.days)\*1) + " rupees only.").grid(row=2, column=0)

confirm1 = Button(frame2, text="Confirm", padx=3, pady=3, command=fineCheck)

confirm1.grid(row=1, column=1)

frame3 = LabelFrame(root, text="Check availability".center(19), padx=6, pady=6)

frame3.grid(row=7, column=0, padx=6, pady=6)

bookLabel = Label(frame3, text="Name: ", anchor=E)

bookLabel.grid(row=0, column=0)

bookEntry = Entry(frame3, width=30, borderwidth=3)

bookEntry.grid(row=0, column=1)

def bookCheck():

    c.execute('SELECT title, quantity FROM books')

    for i, j in c.fetchall():

        if bookEntry.get().lower() == i.lower():

            Label(frame3, text="There are " + str(j) + " copies available.").grid(row=2, column=0)

            break

        else:

            Label(frame3, text="This title is not available.").grid(row=2, column=0)

confirm1 = Button(frame3, text="Confirm", padx=3, pady=3, command=bookCheck)

confirm1.grid(row=1, column=1)

root.mainloop()

c.connection.close()

**4.2 DESCRIPTION OF MODULES:**

**Modules Used:**

1. Tkinter (\*):

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.

1. Datetime (date):

In Python, date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.

Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals.

1. libDatabase (\*):

A database is an abstraction over an operating system’s file system that makes it easier for developers to build applications that create, read, update and delete persistent data. Python supports various databases like SQLite, MySQL, Oracle, Sybase, PostgreSQL, etc.

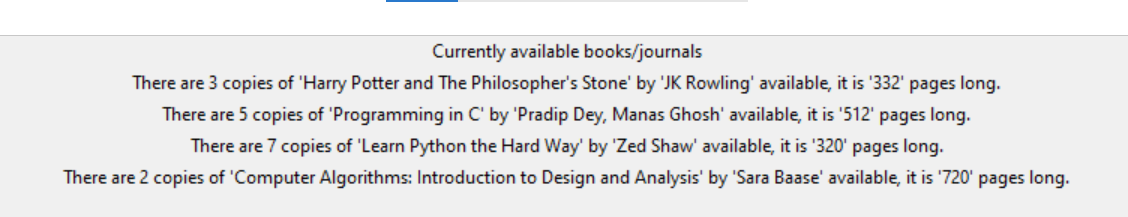
1. SQLite3:

SQLite3 is a very easy to use database engine. It is self-contained, serverless, zero-configuration and transactional. … The Python Standard Library includes a module called “sqlite3" intended for working with this database. This module is a SQL interface compliant with the DB-API 2.0 specification.

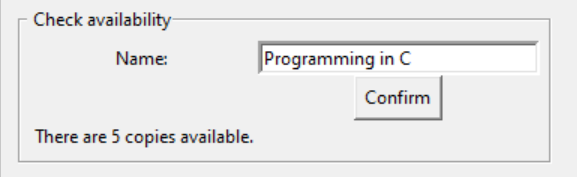
**5.RESULT AND DISCUSSION**:

**5.1 SCREENSHOTS:**

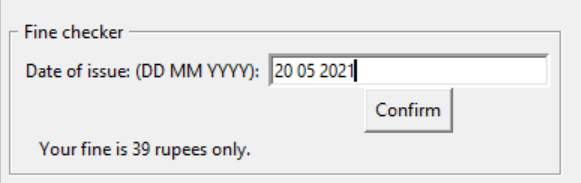
**SOFTWARE INTERFACE:**

**Listing all books/Available books:**

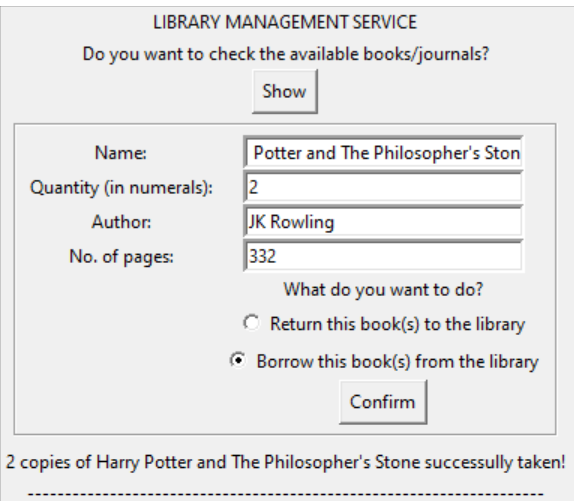
**Search for books:**



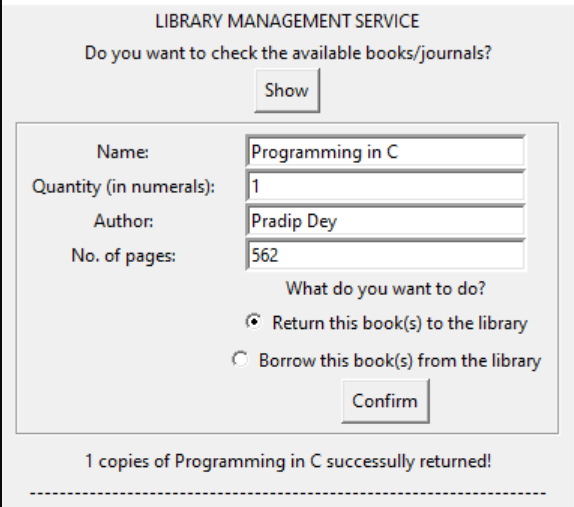
**Fine Checker:**



**Borrowing books:**



**Returning Books:**



**6. CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION:**

* The Library Management System allows the user to store the book details
* This software allows storing the details of all the data related to library.
* The implementation of the system will reduce data entry time and provide readily calculated reports.
  1. **FUTURE ENHANCEMENT:**

**FUTURE SCOPE OF APPLICATION:**

This application can be easily implemented under various situations. We can add new features as we require. There is flexibilityin all the modules.

**SOFTWARE SCOPE:**

**Extensibility:** This software is extendable in ways that its original developers may not expect. The following principles enhances extensibility like hide data structure, avoid traversing multiple links or methods, avoid case statements on object type and distinguish public and private operations.

**Reusability:** Reusability is possible when it isrequired in thisapplication. We can update it to the next version. Reusable softwarereduces design, coding and testing cost by reducing effortover several designs. Reducing the amount of code alsosimplifies understanding, which increases the likelihood thatthe code is correct.

**Understandability:** A method is understandable if someone other than the creator of the method can understand the code (as well as the creator after a time lapse). We use the method, which is small and coherent help to accomplish this.

**Cost-effectiveness:** Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the entire requirement. Scope of this document is to put down the requirements, clearly identifying the information needed by the user, the source of the information and outputs expected from the system.

**REFERENCES**:

.

[1]"ALA | Exhibit Spaces and Bulletin Boards". www.ala.org. Retrieved 2016-09-20.

[2]"Disaster planning". Australian Library and Information Association.

[3]Hawthorne, P (2011). “Succession Planning and Management: A Key Leadership

Responsibility Emerges”. Texas Library Journal. 87 (1): 8–12.

[4]McClure, C. R. (1980). “LIBRARY MANAGERS: Can They Manage? Will They Lead?”

Library Journal: 2391.

[5]Moran, B.; Stueart, R.; Morner, C. (2013). Library and Information Center Management.

California: Libraries Unlimited. ISBN 978-1-59884-989-9.

[6]Pautz, Hartwig. “Income generation in public libraries: potentials and pitfalls”. Library Review.

63 (8/9): 568.

[7] Sharma, C. K.; Singh, Kiran (2005), Library Management, ISBN 978-81-269-0452-5

[8]“The Journal Library Administration”. Retrieved 7 November 2013.

[9]Wellish, Hans H. “Dewey Decimal Classification, Universal Decimal Classification, and the

Broad System of Ordering: The Evolution of Universal Ordering Systems”. College of Library

and Information Sciences, University of Maryland. CiteSeerX 10.1.1.221.3537.