A

MINOR PROJECT REPORT
ON
LIBRARY MANAGEMENT PORTAL

Department Of IBM-ICE
School of Research and Technology, Bhopal (M.P.)
2016-17

People's University, Bhanpur, Bhopal (M.P.)

Submitted To:
Mr. SAMEER HASAN
(Asst.Proff)

Submitted By:
Vivek(PU-018161517A12)
Ruchika (PU-010161517A12)
V.N.S Aditya (PU-016161517A12)
Avinash (PU-010161517A12)

CERTIFICATE

This is to certify that Minor Project entitled “librery Management System” submitted by **vivek kumar** & **Ruchika bele, VNS Aditya** and **avinash Kumar** has been carried out under my guidance/supervision. The Minor project Report is approved for submission towards partial fulfillment for the award of degree of ***Bachelor of Technology*** in Computer Science Engineering.

It is further certified that this work has been submitted for the award of my other degree or diploma.

Mr. Sameer Hasan
Guide/Supervisor
Assistant professor
Computer Science Engineering, People’s College of Research & Technology

Mr. Ankit Dewedi
(*Head of Department*)

School of Research and Technology, IBM- ICE
People’s University, Bhopal

CANDIDATE’S DECLARATION

We hereby declare that the work, which is being presented in the Minor Project Report, entitled "liberery Management System" in partial fulfillment of the requirements for the award of degree Of Bachelor of Technology in Computer Science Engineering, submitted in the Department of Computer Science Engineering, School of Research and Technology, People's University, Bhopal is in authentic record of my own work carried out under the guidance of ***Mr. Sameer Hasan***, Assistant Professor, Department Computer Science Engineering, School of Research and Technology, People's University, Bhopal.

The matter combined in this Minor Project Report has not been submitted by us for the award of any other degree.

Vivek Kumar (PU-018161517A12)
Ruchika Bele (PU010161517A12)
V.N.S Aditya (PU-016161517A12)
Avinash (PU-026161517A12)

CERTIFICATE

This is certify that above statement made by the candidate is correct to the best of my knowledge.

Mr. Sameer Hasan
Guide/Supervisor
Assistant professor
Department of Computer Science Engineering

School of College of Research & Technology, IBM-ICE
People’s University, Bhopal

ACKNOWLEDGEMENT

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, **Mr. SAMEER HASAN**, for providing us with the right guidance and advice at the crucial junctures and for showing me the right way. We also take this opportunity to express a deep sense of gratitude to our class coordinator **Ms. RADHA RASHMI TIWARI** for their cordial support, valuable suggestions and guidance. We extend our sincere thanks to our respected **Head of the division Mr. ANKIT DWIVEDI**, for allowing us to use the facilities available. We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank our friends and family for the support and encouragement they have given us during the course of our work.

TABLE OF CONTENTS

ABSTRACT

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like facility of user login and a facility of teachers login .It also has a facility of admin login through which the admin can monitor the whole system .It also has facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board . It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date and also the students can request the librarian to add new books by filling the book request form. The librarian after logging into his account i.e. admin account can generate various reports such as student report, issue report, teacher report and book report

Overall this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

CHAPTER 1

INTRODUCTION

This chapter gives an overview about the aim, objectives, background and operation environment of the system.

1.1 PROJECT AIMS AND OBJECTIVES

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- Online book issue
- Request column for librarian for providing new books
- Student login page where student can find books issued by him/her and date of return.
- A search column to search availability of books
- Notification to student for return book before due date.

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 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.

1.3 OPERATION ENVIRONMENT

PROCESSOR	INTEL CORE PROCESSOR OR BETTER PERFORMANCE
OPERATING SYSTEM	WINDOWS 10 ,WINDOWS7, UBUNTU
MEMORY	4GB RAM OR MORE
HARD DISK SPACE	MINIMUM 3 GB FOR DATABASE USAGE FOR FUTURE
DATABASE	MY SQL

CHAPTER 2

SYSTEM ANALYSIS

In this chapter, we will discuss and analyze about the developing process of Library Management System including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

2.1 SOFTWARE REQUIREMENT SPECIFICATION

2.1.1 GENERAL DESCRIPTION

PRODUCT DESCRIPTION:

Library Management System is a computerized system which helps User (librarian) to manage the library daily activity in electronic format. It reduces The risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and time-Saving.

PROBLEM STATEMENT:

The problem occurred before having computerized system includes:

- File lost

  -When computerized system is not implemented file is always lost because of human Environment. Sometimes due to some human error there may be a loss of records.

- File damaged

  -When a computerized system is not there file is always lost due to some accident like spilling of water by some member on file accidentally Besides some natural Disaster like floods or fires may also damage the files.

- Difficult to search record

  -When there is no computerized system there is always a difficulty in searching of Records if the records are large in number.

- Space consuming
  -After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

- Cost consuming
  -As there is no computerized system to add each record paper will be needed which will increase the cost for the management of library.

2.1.2 SYSTEM OBJECTIVES

- Improvement in control and performance

  -The system is developed to cope up with the current issues and problems of library .The system can add user, validate user and is also bug free.

- Save cost

  -After computerized system is implemented less human force will be required to Maintain the library thus reducing the overall cost.

- Save time

  -Librarian is able to search record by using few clicks of mouse and few search Keywords thus saving his valuable time.

- Option of online Notice board

  -Librarian will be able to provide a detailed description of workshops going in the college as well as in nearby colleges

- Lecture Notes

- Teacher have a facility to upload lectures notes in a pdf file having size not more than 10mb

2.1.3 SYSTEM REQUIREMENTS

2.1.3.1 NON FUNCTIONAL REQUIREMENTS

- **Product Requirements**

EFFICIENCY REQUIREMENT

When a library management system will be implemented librarian and user will easily access library as searching and book transaction will be very faster.

RELIABILITY REQUIREMENT

The system should accurately performs member registration, member validation, report generation, book transaction and search

USABILITY REQUIREMENT

The system is designed for a user friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

ORGANIZATIONAL REQUIREMENT

IMPLEMENTATION REQUIREMNTS

In implementing whole system it uses PYTHON in front end with MYSQL as server side scripting language which will be used for database connectivity and the backend i.e. the database part is developed using MySQL.

DELIVERY REQUIREMENTS

The whole system is expected to be delivered in six months of time with a weekly evaluation by the project guide.

2.1.3.2 FUNCTIONAL REQUIREMENTS

1. NORMAL USER

1.1 USER LOGIN

Description of feature

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system .The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

Functional requirements

-user id is provided when they register

-The system must only allow user with valid id and password to enter the system

-The system performs authorization process which decides what user level can acess to.

-The user must be able to logout after they finished using system.

1.2 REGISTER NEW USER

Description of feature

This feature can be performed by all users to register new user to create account.

Functional requirements

-System must be able to verify information

-System must be able to delete information if information is wrong

1.3 REGISTER NEW BOOK

Description of feature

This feature allows to add new books to the library

Functional requirements

-System must be able to verify information

-System must be able to enter number of copies into table.

- System must be able to not allow two books having same book id.

1.4 SEARCH BOOK

DESCRIPTION OF FEATURE

This feature is found in book maintenance part. We can search book based on book ID, book name publication or by author name.

Functional requirements

- System must be able to search the database based on select search type
- System must be able to filter book based on keyword entered
- System must be able to show the filtered book in table view

1.5 ISSUE BOOKS AND RETURN BOOKS

DESCRIPTION OF FEATURE

This feature allows to issue and return books and also view reports of book issued.

Functional requirements

-System must be able to enter issue information in database.

-System must be able to update number of books.

- System must be able to search if book is available or not before issuing Books.

-System should be able to enter issue and return date information.
-System should be able to notify student for their due date.

-System should be able to add detailed information about student.

2.1.4 SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system

2.1.4.1 SOFTWARE REQUIREMENTS

- Operating system- Windows 10 is used as the operating system as it is stable and Supports more features and is more user friendly
- Database MYSQL-MYSQL is used as database as it easy to maintain and retrieve records by simple queries which are in English language which are easy to Understand and easy to write.
- Development tools and Programming language- PYTHON is used to write the whole.

2.1.4.2 HARDWARE REQUIREMENTS

Intel core i3 2nd generation is used as a processor because it is fast than other processors and provide reliable and stable and we can run our pc for longtime. By using this processor we can keep on developing our project without any worries.

Ram 4 GB is used as it will provide fast reading and writing capabilities and will in turn support in processing

2.2 EXISTING VS PROPOSED SYSTEM

i. Existing system does not have any facility of teachers login or student login whereas proposed system will have a facility of student login as well as teacher’s login

ii. Existing system does not have a facility of online reservation of books whereas proposed system has a facility of online reservation of books

iii. Existing system does not have any facility to generate student reports as well book issue reports whereas proposed system provides librarian with a tool to generate reports

iv. Existing system does not has any facility for book request and suggestions where as in proposed system after logging in to their accounts student can request books as well as provide suggestions to improve library

v. Existing system does not have any option for near due date notification.

vi. Features of audit and trails.

2.3 SOFTWARE TOOLS USED

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

2.3.1 Front end

The front end is designed using of PYTHON.

PYTHON-

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

MYSQL-

MySQL ("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Wideners daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single

For-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation

.MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, Word Press, phpDB, MyDB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

CHAPTER 3

SYSTEM DESIGN

3.1 TABLE DESIGN

VARIOUS TABELS TO MAINTAIN INFORMATION

BOOK TABLE FOR KEEPING TRACK OF BOOKS

Field	Data type	Default	Key	Extra
BookID	INT(11)	Not Null	identity	Auto increment
Title	VARCHAR(255)	Null		
Author	VARCHAR(255)	Null		
Subject	VARCHAR(255)	Null		
Available copies	INT(10)	Null		
Total Copies	INT(10)			

STUDENT TABLE FOR STUDENT INFORMATION

Field	Data type	Default	Key	Extra
StudentID	INT(11)	NOT NULL	Primary key	
name	INT(10)	NULL		
year	VARCHAR(255)	NULL		
contact	VARCHAR(255)	NULL		
email	VARCHAR(255)	NULL		

ORROW TABLE TO KEEP TRACK OF BOOK ISSUIED

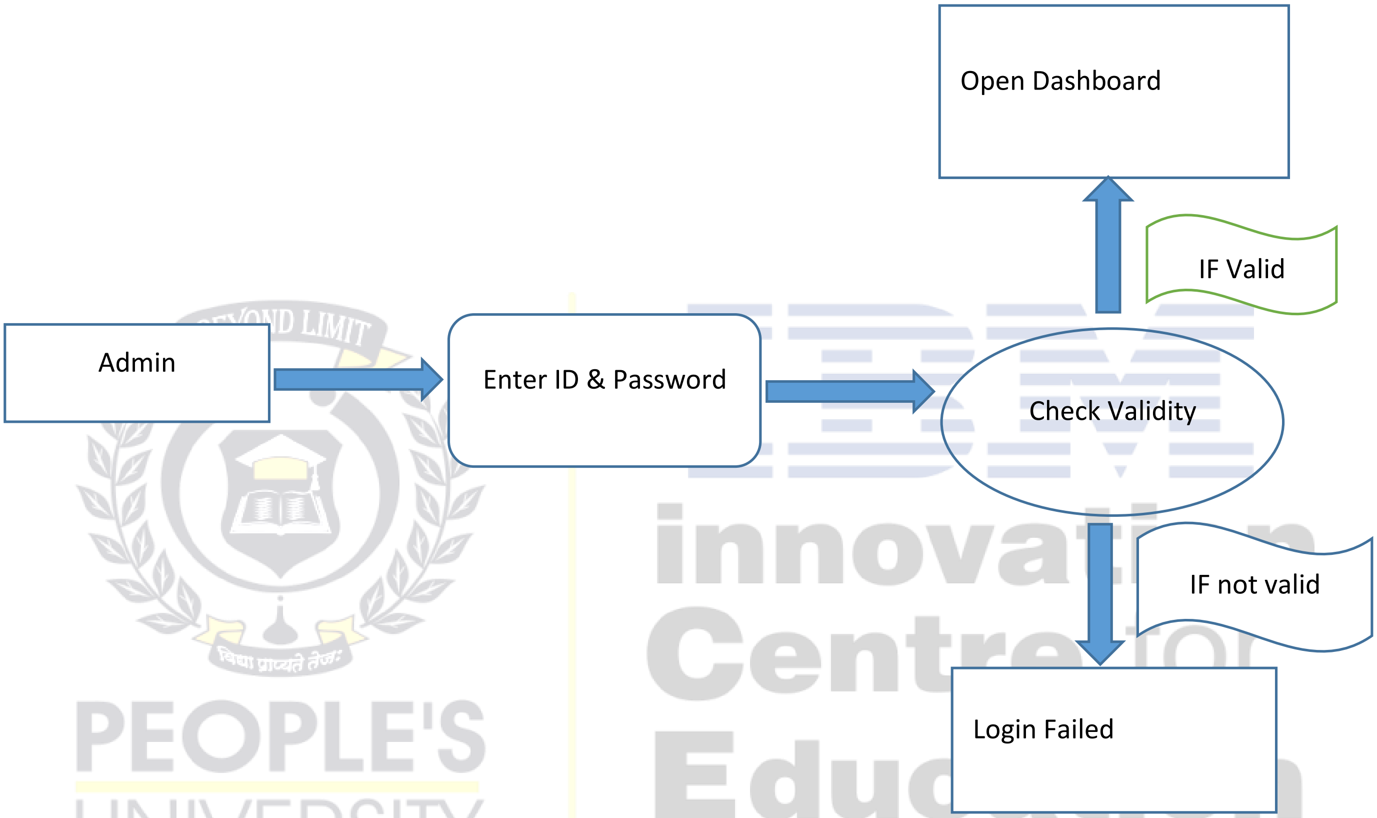
Field	Data Type	Default	Key	Extra
BorrowID	INT(11)	NOT NULL	Primary key	
StudentID	INT(11)	NOT NULL	Foreign key	References Student_tbl (StudentID)
BookID	INT	NULL	Foreign key	References Book_tbl (BookID)
Duedate	DATE	NULL		
ReturnDate	DATE	NULL		
Notes	nvarchar(50)			
Status	nvarchar(50)			

STUDENT LOGIN TABLE

Field	Data type	Default	Key	Extra
loginid	INT(11)	NOT NULL	Foreign key	References Student
Username	VARCHAR(255)	NULL		
Password	VARCHAR(255)	NULL		

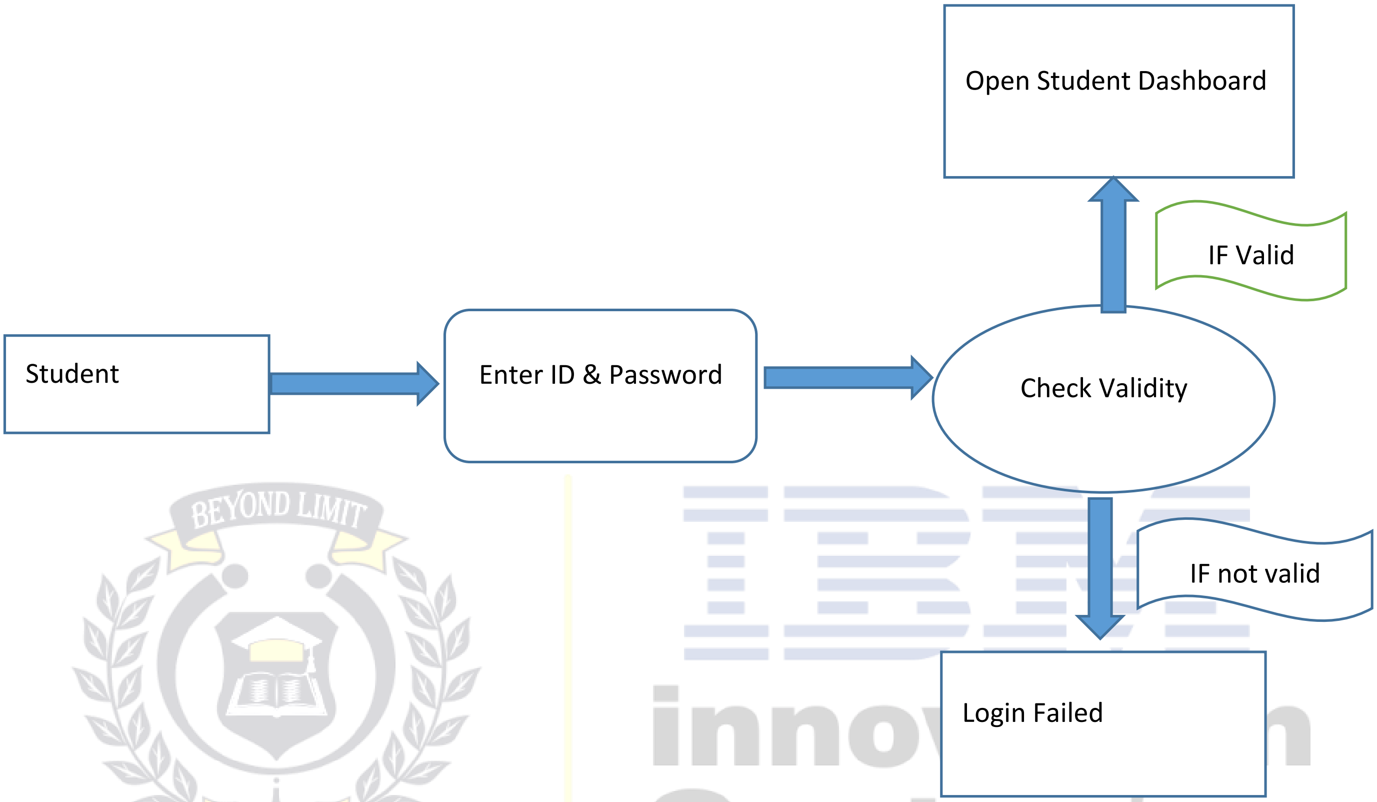
3.2 DATA FLOW DIAGRAMS

DATA FLOW DIAGRAM FOR ADMIN LOGIN



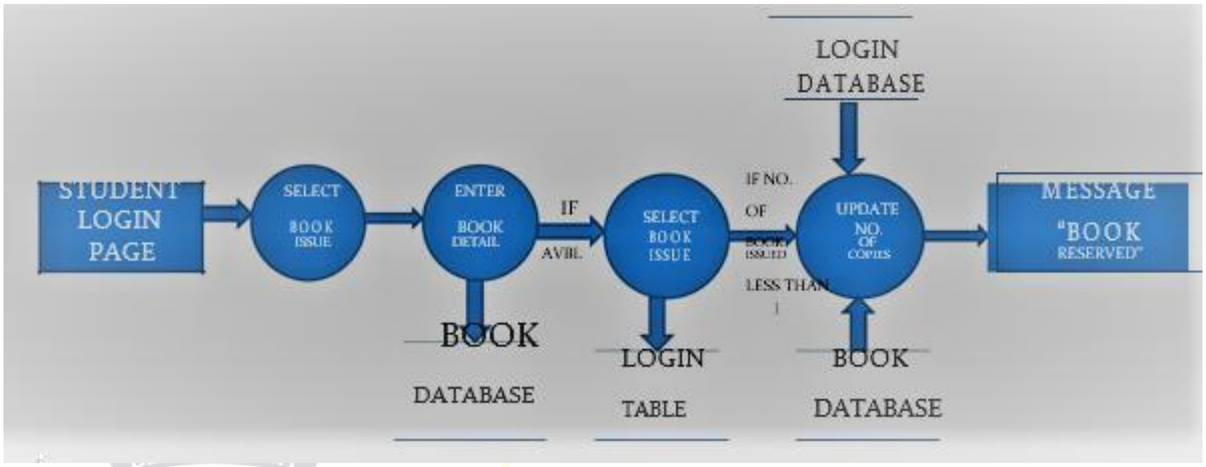
After entering to the home page of the Panel, Admin can choose the Admin LOGIN option where they are asked to enter username & password, and if he/she is a valid user then a Admin login page will be displayed.

DATA FLOW DIAGRAM FOR STUDENT LOGIN



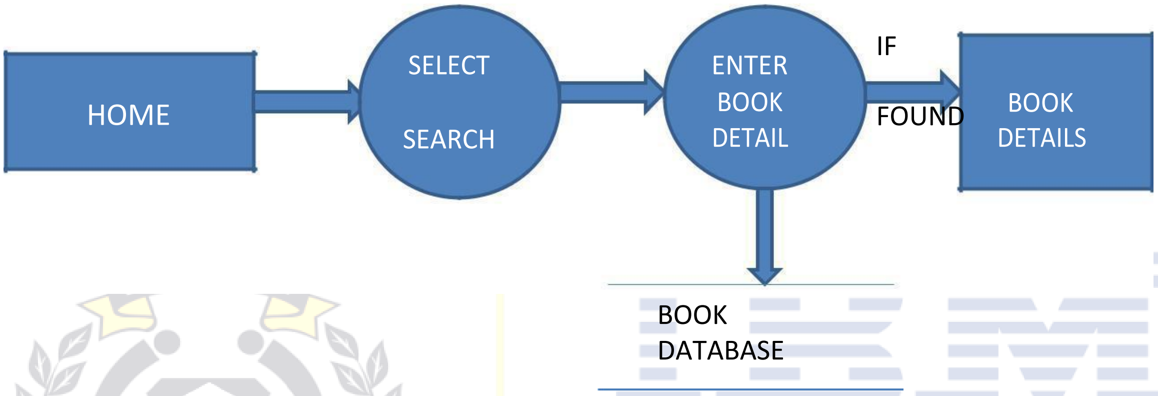
After entering to the home page of the portal, student can choose the STUDENT LOGIN option where they are asked to enter username & password, and if he/she is a valid user then a student login page will be displayed.

DATA FLOW DIAGRAM FOR BOOK ISSUE



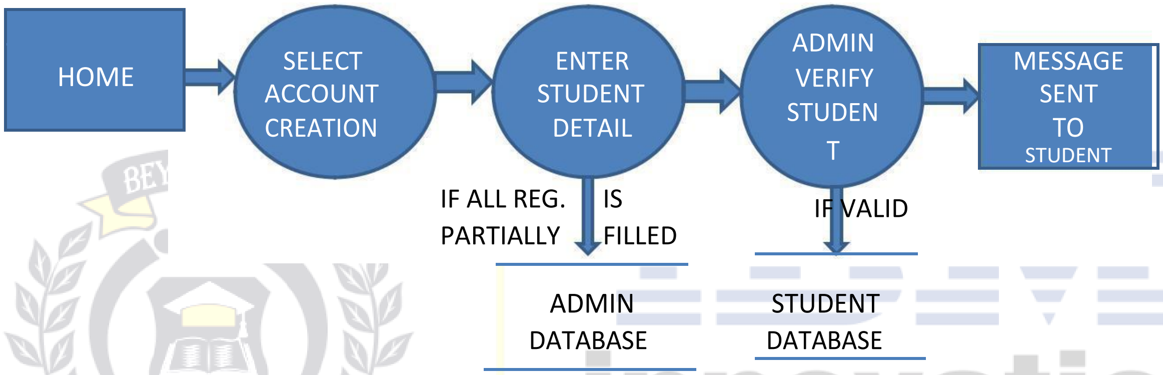
It is a 2nd level Data Flow Diagram where after entering STUDENT LOGIN page he/she can select a book issue option where after entering the book detail, he/she can select the book issue option and if the maximum no of books issued limit is not crossed then a request will be sent to the librarian who will approve the book issue.

DATA FLOW DIAGRAM FOR BOOK SEARCH



After the home page login there will be an option of the book search where after entering book detail like author name, publication, book name etc. book details will be displayed.

DATA FLOW DIAGRAM FOR ACCOUNT CREATION



After the home page login there will be an option of CREATE AN ACCOUNT where after entering student detail, if all the fields are filled then a request will be sent to the librarian who will approve him as a registered member of the library.

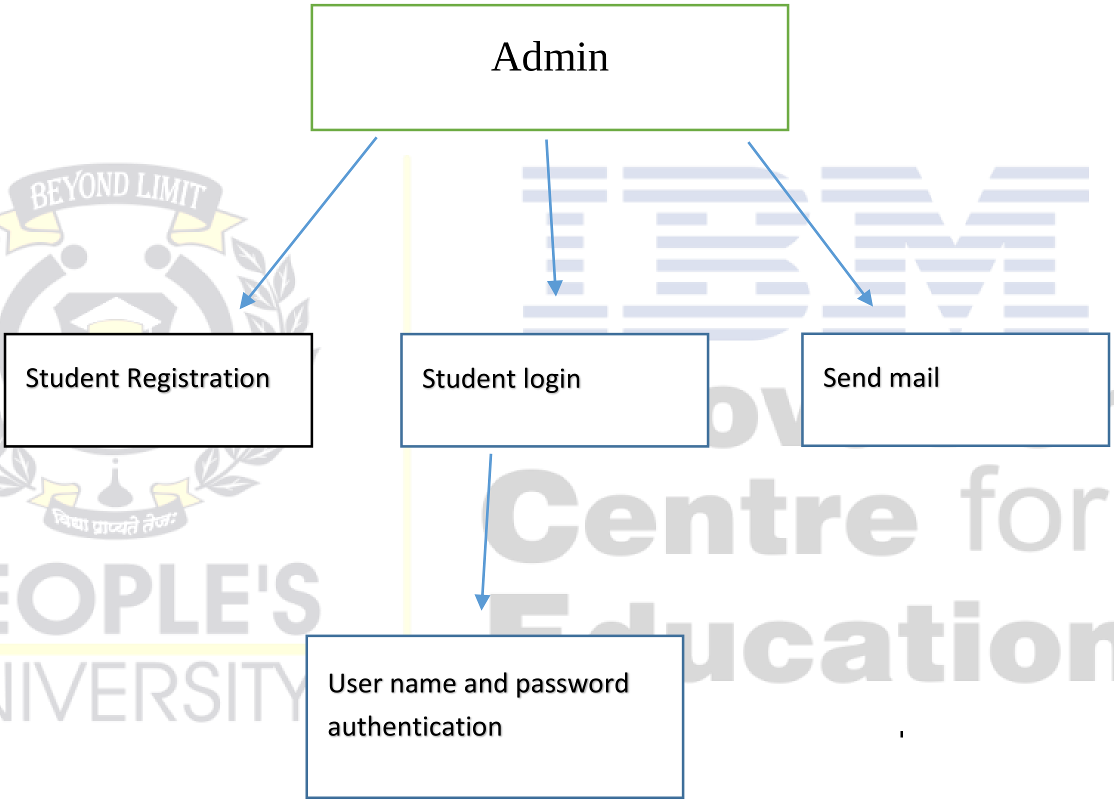
CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION

For Library Management System it is divided into the following Modules:

4.1.2 Admin Module



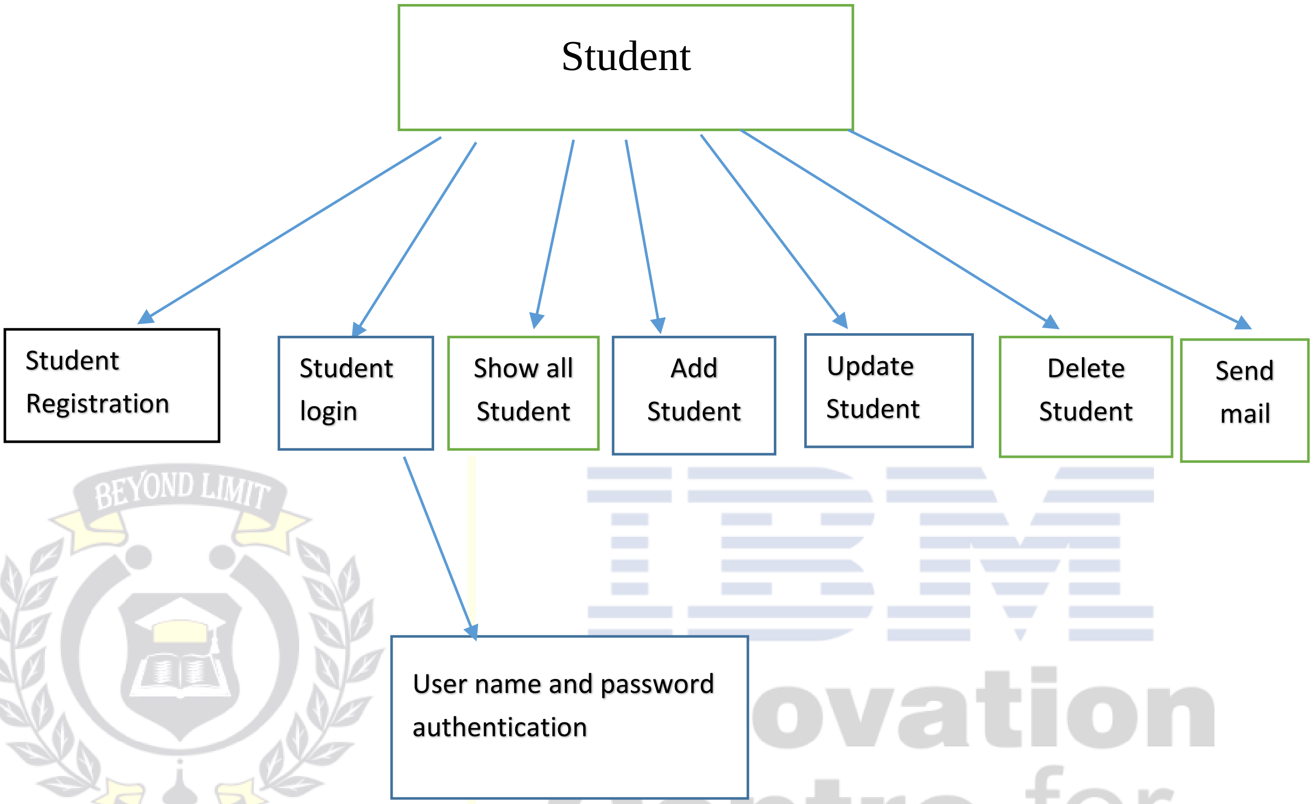
The following module contains various facilities like student validation, student registration, book addition, and report generation.

4.1.1.1 Code for Admin module

4.1.1.1 Code for entering admin username and password

```
from tkinter import *
import pymysql
from tkinter import messagebox as ms
from tkinter import messagebox


 def loginn():

    a = nameE.get()
    b = passwdE.get()

    print(a)
    print(b)

    db = pymysql.connect(host="localhost", user="root", password="redhat", db="test")
    cursor = db.cursor()

    # find_user = ("SELECT * FROM test.login WHERE username = '"+a+"' and
passwords = "+b+" ")

    find_user = ("SELECT * FROM test.login WHERE username = 'vivek' and passwords =
123 ")

    cursor.execute(find_user)
    result = cursor.fetchone()
```

4.1.2Student Module



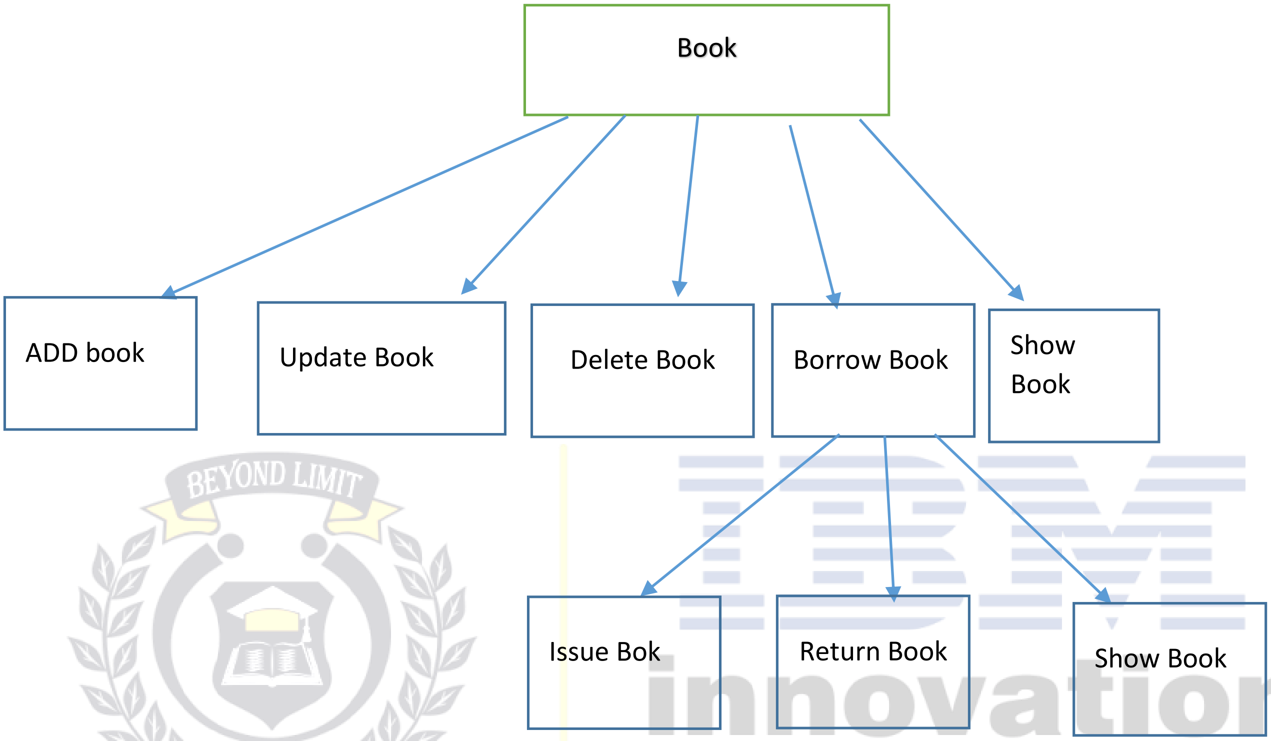
The following module contains various facilities like student registration, student login, online book reservation, and event addition. Any student if at any moment forgets his password he can retrieve it from forgot password option.

4.1.2.1 Code For Student registration

```
def stu():

         sup = Tk()
         sup.resizable(width=FALSE, height=FALSE)
         sup.geometry('900x700')
         sup.configure(bg='dark gray')
         sup.title('Sign up')

lblwel = Label(sup, text="ADD NEW STUDENT RECORD", font=("bold", 20)).place(x=300,
y=30)
fnamesup = Label(sup, text="First Name", font=("bold", 15)).place(x=300, y=100)

lnamesup = Label(sup, text="Last Name", font=("bold", 15)).place(x=300, y=160)

dptsup = Label(sup, text="Department Name", font=("bold", 15)).place(x=300, y=220)
```

```
branchsup = Label(sup, text="Branch", font=("bold", 15)).place(x=300, y=280)

semestersup = Label(sup, text="Semester", font=("bold", 15)).place(x=300, y=340)

yearsup = Label(sup, text="Year", font=("bold", 15)).place(x=300, y=400)

Enrollsup = Label(sup, text="En.Roll", font=("bold", 15)).place(x=300, y=460)

e1 = Entry(sup, width=20, font=("bold", 15), highlightthickness=2, bg="powder blue",
relief=SUNKEN)
e1.place(x=500, y=100)
e2 = Entry(sup, width=20, font=("bold", 15), show="*", highlightthickness=2,
bg="powder blue",
relief=SUNKEN)
e2.place(x=500, y=160)
e3 = Entry(sup, width=20, font=("bold", 15), highlightthickness=2, bg="powder blue",
relief=SUNKEN)
e3.place(x=500, y=220)
# default = StringVar()
e4 = Entry(sup, width=20, font=("bold", 15), highlightthickness=2, bg="powder blue",
relief=SUNKEN)
##default.set("root")
e4.place(x=500, y=280)
e5 = Entry(sup, width=20, font=("bold", 15), highlightthickness=2, bg="powder blue",
relief=SUNKEN)
e5.place(x=500, y=340)

e6 = Entry(sup, width=20, font=("bold", 15), highlightthickness=2, bg="powder blue",
relief=SUNKEN)
e6.place(x=500, y=400)

e7 = Entry(sup, width=20, font=("bold", 15), highlightthickness=2, bg="powder blue",
relief=SUNKEN)
e7.place(x=500, y=460)
btnadd = Button(sup, padx=8, bd=8, fg='black', font=('arial', 10, 'bold'),
text='ADD')#, command=add)
btnadd.place(x=400, y=550)
btncancelsup = Button(sup, padx=8, bd=8, fg='black', font=('arial', 10, 'bold'),
text='Quit',
command=quit)
btncancelsup.place(x=550, y=550)
# b1 = Button(sup, text="Sign Up", width=25, height=2, bg=g, fg="white", font="5",
relief=RAISED, overrelief=RIDG
```

4.1.4 Book Module



The following module contains various facilities like add new book and search book. In the 'add new book' section if any new book comes in the library then the librarian can add its specifications. Similarly if the user wants to search for a specific book then he/she can use search book option to do it.

4.1.4.1 Code for Adding New books

```
def book():

        def addbook():

                 def btnadd():
                 b= bidE.get()
                ba = btitleE.get()
                 bb = bauthorE.get()
                bc=byearE.get()
                bd=bsubjectE.get()
                be=bavicopiesE.get()
                bf=btotalcopiesE.get()

                 db = pymysql.connect(host="localhost", user="root",
password="redhat", db="test")
                cursor = db.cursor()

                 ins_book= ("INSERT  INTO
                test.book_tbl(BookId,Title,Author,Year,Subject,AvailableCopies,T
                otalCopies) VALUES("+b+",' " + ba + " ',' " + bb + " ', " + bc + " ,'
                " + bd + " '," + be + "," + bf + ");")

                 cursor.execute(ins_book)

                 ms._show("Book ADD", "Successfully Added")

                db.commit()
                db.close()


            if result:
              print("successfull")

            else:
              print ("ERROR")
```

4.1.4.2 Code for Book Search

```
def search():
            try:
              db = pymysql.connect(host="localhost", user="root",
password="redhat", db="test")
              cursor = db.cursor()
              find_user = "select * from test.book_tbl where BookId = '%s'" %
idbook.get()
```

```
        # find_user = "select * from test.book_tbl where BookId = 1"
        cursor.execute(find_user)
        result = cursor.fetchone()
        Title.set(result[1])
        Author.set(result[2])
        Year.set(result[3])
        Subject.set(result[4])
        AvailableCopies.set(result[5])
        TotalCopies.set(result[6])
        e1.configure(state='disabled')
        db.commit()
        db.close()
    except:
        messagebox.showinfo('no data')
        clear()
```

4.1.5.1 Code for issue book

```
def issuebook():
    roots = Tk()

    roots.title("ISSUE BOOK")

    roots.geometry("255x450")
    roots.configure(bg='#ffb900')

    StuidL = Label(roots, text="Student Id", )
    StuidL.place(x=20, y=40)

    StuidE = Entry(roots, bg="powder blue")
    StuidE.place(x=100, y=40)

    BookIdL = Label(roots, text="Book Id", )
    BookIdL.place(x=20, y=90)

    BookIdE = Entry(roots, bg="powder blue")
    BookIdE.place(x=100, y=90)

    BorrowDL = Label(roots, text="Borrow Date", )
    BorrowDL.place(x=20, y=140)

    BorrowDE = Entry(roots, bg="powder blue")
    BorrowDE.place(x=100, y=140)

    DuedateL = Label(roots, text="Due Date", )
    DuedateL.place(x=20, y=190)
```

```
                DuedateE = Entry(roots, bg="powder blue")
                DuedateE.place(x=100, y=190)

                ReturndateL = Label(roots, text="Return Date", )
                ReturndateL.place(x=20, y=240)

                ReturndateE = Entry(roots, bg="powder blue")
                ReturndateE.place(x=100, y=240)

                Notesl = Label(roots, text="Notes", )
                Notesl.place(x=20, y=290)

                NotesE = Entry(roots, bg="powder blue")
                NotesE.place(x=100, y=290)

                Statusl = Label(roots, text="Notes", )
                Statusl.place(x=20, y=340)

                StatusE = Entry(roots, bg="powder blue")
                StatusE.place(x=100, y=340)

                btnissue = Button(roots, padx=8, bd=8, fg='black', font=('arial', 10,
      'bold'), text='Issue')
                btnissue.place(x=40, y=400)

                btncancel = Button(roots, padx=8, bd=8, fg='black', font=('arial', 10,
      'bold'), text='Quit')
                btncancel.place(x=150, y=400)
                roots.mainloop()
```

4.1.5.2 Code for book Return

```
def returnbook():
            roots = Tk()

            roots.title("LOGIN")

            roots.geometry("255x450")
            roots.configure(bg='#ffb900')

            StuidL = Label(roots, text="Student Id", )
            StuidL.place(x=20, y=40)

            StuidE = Entry(roots, bg="powder blue")
            StuidE.place(x=100, y=40)

            BookIdL = Label(roots, text="Book Id", )
```

```
BookIdL.place(x=20, y=90)

BookIdE = Entry(roots, bg="powder blue")
BookIdE.place(x=100, y=90)

BorrowDL = Label(roots, text="Borrow Date", )
BorrowDL.place(x=20, y=140)

BorrowDE = Entry(roots, bg="powder blue")
BorrowDE.place(x=100, y=140)

DuedateL = Label(roots, text="Due Date", )
DuedateL.place(x=20, y=190)

DuedateE = Entry(roots, bg="powder blue")
DuedateE.place(x=100, y=190)

ReturndateL = Label(roots, text="Return Date", )
ReturndateL.place(x=20, y=240)

ReturndateE = Entry(roots, bg="powder blue")
ReturndateE.place(x=100, y=240)

Notesl = Label(roots, text="Notes", )
Notesl.place(x=20, y=290)

NotesE = Entry(roots, bg="powder blue")
NotesE.place(x=100, y=290)

Statusl = Label(roots, text="Notes", )
Statusl.place(x=20, y=340)

StatusE = Entry(roots, bg="powder blue")
StatusE.place(x=100, y=340)

btnissue = Button(roots, padx=8, bd=8, fg='black', font=('arial', 10, 'bold'), text='Issue')
btnissue.place(x=40, y=400)

btncancel = Button(roots, padx=8, bd=8, fg='black', font=('arial', 10, 'bold'), text='Quit')
btncancel.place(x=150, y=400)
roots.mainloop()
```

CHAPTER 5

SYSTEM TESTING

The aim of the system testing process was to determine all defects in our project .The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not.
Our Project went through two levels of testing

1 .Unit testing
2. Integration testing

UNIT TESTING

Unit testing is undertaken when a module has been created and successfully reviewed .In order to test a single module we need to provide a complete environment i.e. besides the module we would require

- The procedures belonging to other modules that the module under test calls
- Non local data structures that module accesses
- A procedure to call the functions of the module under test with appropriate parameters

Unit testing was done on each and every module that is described under module description of chapter 4

1. Test for the admin module

- Testing admin login form-This form is used for log in of administrator of the system. In this we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password
- Student account addition- In this section the admin can verify student details from student academic info and then only add student details to main library database it contains add and delete buttons if user click add button data will be added to student database and if he clicks delete button the student data will be deleted
- Book Addition- Admin can enter details of book and can add the details to the main book Table also he can view the books requests.

2. Test for Student login module

- Test for Student login Form-This form is used for log in of Student .In this we enter the library ID, username and password if all these are correct student login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for library ID, username and password.
- Test for account creation- This form is used for new account creation when student does not fill the form completely it asks again to fill the whole form when he fill the form fully it gets redirected to page which show waiting for conformation message as his data will be only added by administrator after verification.

3. Test for teacher login module-

Test for teacher login form- This form is used for log in of teacher .In this we enter the username and password if all these are correct teacher login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password.

INTEGRATION TESTING

In this type of testing we test various integration of the project module by providing the input .The primary objective is to test the module interfaces in order to ensure that no errors are Occurring when one module invokes the other module

CHAPTER 6

CONCLUSION & FUTURE SCOPE

This website provides a computerized version of library management system which will benefit the students as well as the staff of the library.

It makes entire process online where student can search books, staff can generate reports and do book transactions. It also has a facility for student login where student can login and can see status of books issued as well request for book or give some suggestions.

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility, a feature Of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfills each users need in the best way possible.

CHAPTER 7

REFERENCES

- http://www.w3schools.com/html/html_intro.asp
- http://www.w3schools.com/js/js_datatypes.asp
- http://www.w3schools.com/sql/sql_insert.asp
- http://www.w3schools.com/sql/sql_update.asp
- http://www.w3schools.com/php/php_forms.asp
- www.youtube.com
- www.python.org