**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANA SANGAMA, BELAGAVI – 590018.**



**MINI PROJECT REPORT**

**ON**

**“LIBRARY MANAGEMENT SYSTEM”**

Submitted in partial fulfillment for the requirement of 5th semester for the

**Degree of Bachelor of Engineering in**

**COMPUTER SCIENCE & ENGINEERING**

For the academic year 2020-21

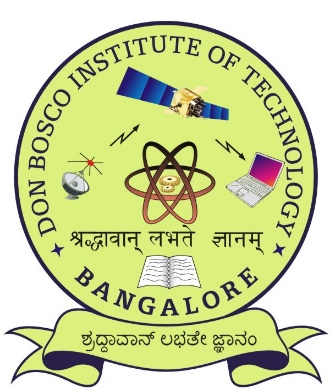
**SUBMITTED BY:**

**SANDESH TIWARI S (1DB18CS118)**

Under the guidance of

**Mr. Giridhar Gowda**

Assistant Professor, Dept. of CSE

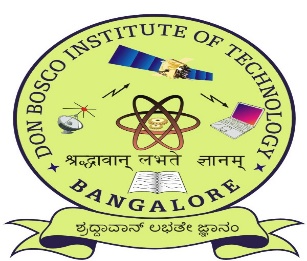


**DON BOSCO INSTITUTE OF TECHNOLOGY, BANGALORE-560074**

**DON BOSCO INSTITUTE OF TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Kumbalagodu, Bangalore-560074**

****

**CERTIFICATE**

This is to certify that the Mini Project Report entitled **“LIBRARY MANAGEMENT SYSTEM”** is a bonafide Mini Project work carried out by **SANDESH TIWARI S (1DB18CS118)**, in partial fulfillment of ‘5th’ semester for the Degree of **Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi, during the academic year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated with the degree mentioned.

Signature of Guide Signature of HOD

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**   **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Giridhar Gowda Prof. B.S. Umashankar** Assistant Professor, Professor and Head,

Dept. of CSE, Dept. of CSE,

DBIT, Bangalore. DBIT, Bangalore.

**External Viva**

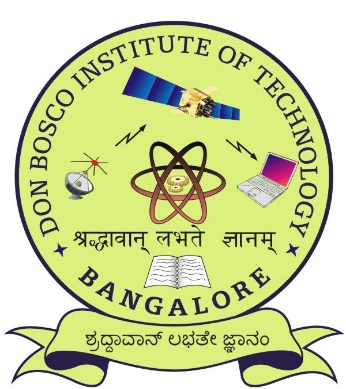
**Name of the Examiners Signature with Date**

1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**DON BOSCO INSTITUTE OF TECHNOLOGY**

**Kumbalagodu, Bangalore-560074**

****

**DECLARATION**

I, **SANDESH TIWARI S**, student of fifth semester B.E, Department of Computer Science and Engineering, Don Bosco Institute of Technology, Kumbalagodu, Bangalore, declare, that the Mini Project Work entitled **“LIBRARY MANAGEMENT SYSTEM”** has been carried out by and submitted in partial fulfillment of the requirement of V semester Aug 2020 - Jan 2021. The matter embodied in this report has been submitted to any university or institute for the award of any other degree or diploma.

**Place:** Bangalore  **SANDESH TIWARI S (1DB18CS118)**

**Date:**

**ACKNOWLEDGEMENT**

At the various stages in making the mini project, a number of people have given me invaluable comment on the manuscript. I take this opportunity to express my deepest gratitude and appreciation to all those who helped me directly or indirectly towards the successful completion of this project.

I would like to thank **Principal Dr. HEMADRI NAIDU T, Don Bosco Institute of Technology** for his support throughout this project.

I express my whole hearted gratitude to **Prof. B.S. Umashankar,** who is our respectable **Head of Dept. of Computer Science**. I wish to acknowledge for his valuable help and encouragement.

In this regard I owe a heartfelt gratitude to my guide **Mr. Giridhar Gowda, Asst.Professor of Department of Computer Science and Engineering**, for his timely advice on the mini project and regular assistance throughout the project work. I would also like to thank the staff members of Department of Computer Science and Engineering for their corporation.

**ABSTRACT**

The main aim and objective were to plan and program system application. We justify to apply the best software engineering practice for system application. I developed a **“LIBRARY MANAGEMENT SYSTEM”** using Java and SQL.

This project provides a platform to store the details of publisher, book, library branches and members borrowing books. And in dynamic way of accessing the data of issued books and returned books, maintaining in real time basis and upgrading the information of all books borrowed and returned.

To know the information of issued and returned books, it is necessary to have a database used to store all these details. Since to obtain the above statement and updating in real time of all issued and returned records can be done through this mini project. This project also reduces the work of manually storing details and messing up things. And the tools used to design and develop this useful project, we use MySQL database as backend and Netbeans as frontend. The main intension of library management is to record the details of issued and returned books in library.

**CONTENTS**

**CHAPTERS PAGE NO**

**1. INTRODUCTION 1-2**

**1.1 Aim**

**1.2 Objective**

**1.3 Scope**

**1.4 Advantages/Disadvantages**

**1.5 Technology Used**

**1.6 System Requirements**

**2. OBJECTIVES 3**

**3. ER DIAGRAM 4**

**4. SCHEMA DIAGRAM 5**

**5. DATA TABLES 6-7**

**6. SOURCE CODES 8-12**

**6.1 Importing packages**

**6.2 Insert statements**

**6.3 Delete statements**

**6.4 Update statements**

**6.5 Display statements**

**7. SNAPSHOTS 13-16**

**8. CONCLUSION 17**

**9. REFERENCES 18**

**Chapter-1**

**INTRODUCTION**

I have developed Library Management System to get rid from manual entry and record system and try to give easy and simple database management system to maintain the record of book. This is generally used by Admin. This system is useful in keeping the record of the publishers, books, library branches, card details and issued and returned details. Since the information is maintained by a single person (Admin), hence the data is safe. By using this System, we can make the system better and quick.

**1.1 Aim**

The main aim of designing this project is to get rid from manual entry and record system and try to give easy and simple database management system for Library.

**1.2 Objective**

The main objective of this DBMS mini project is to construct good quality and

dynamic management system, in which this database is used to store the details of

all the borrowed books.

**1.3 Scope**

The software product “**Library Management System**” will be an application that will be Used for maintaining the records in an organized manner and to replace old paper work system. This project aims at automating the members details for smooth working of the database by automating almost all the activities. Updations and modifications will be easily achievable.

**1.4 ADVANTAGES/DISADVANTAGES**

**1.4.1 Advantages**

This Project is beneficial for the following

1. User has complete control as it provides and accepts only appropriate and valid data.

2. Addition, deletion, modification and display of records as when needed.

3. Decreases the paper and labor work.

4. Manage the entire process.

5. User-friendly error messages are provided wherever necessary.

**1.4.2 Disadvantages**

1. It’s too tiring to give Computerized Timing.

2. Security Limitations.

3. Only a single user (Admin) can access the data.

**1.5 Technology Used: -**

Language : - Java

Backend : - MySQL 8.0

Frontend : - Apache NetBeans IDE 12.1

Drivers : - MySQL connector 8.0.22

**1.6 System Requirements: -**

Minimum RAM : - 4 GB

Hard Disk : - 40 GB

Processor : - Intel Core i5

Operating System : - Windows 10

**Chapter-2**

**OBJECTIVES**

* The Admin has complete control as it provides and accepts only appropriate and valid data.
* The basic purpose of designing this mini project is to get rid from manual entry and record system and try to give easy and simple database management system for Library Management.
* This mini project is designed to keep the record of the Publisher, Book, Library branch, Card details and borrowing details.
* It also helps to keep record of these details and enable an interactive way of maintaining and analyzing the data.

**Chapter-3**

**ER DIAGRAM**

Library Branch

Book

N

M N

Issue

Published by

N

Card

1

Publisher

**ER DIAGRAM of Library Management System**

The figure shows the representation of ER diagram of Library Management System. It contains the connection i.e., relation between the entities and the participation ratio. And primary key is underlined as we see in figure and foreign keys are the keys that relate to primary key of another table represented by connecting to that table.

**Chapter-4**

**SCHEMA DIAGRAM**

Publisher

|  |  |  |
| --- | --- | --- |
| Name | Phone | Address |

Book

|  |  |  |  |
| --- | --- | --- | --- |
| Book\_id | Title | Pub\_year | Publisher\_name |

Library\_branch

|  |  |  |
| --- | --- | --- |
| Branch\_id | Branch\_name | Address |

Card

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Card\_no | Name | Age | Phone | Address | Occupation | Gender |

Issue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Book\_id | Branch\_id | Card\_no | Date\_out | Due\_date |

**SCHEMA DIAGRAM of Library Management System**

The figure shows the representation of Schema diagram of Library Management System. It contains all the tables used in this mini project and these tables are connected to each other with respect to primary keys and foreign keys. Here primary keys are represented by underlining it and foreign keys are connected to the table of that particular primary key is present.

**Chapter-5**

**DATA TABLES**

**Publisher Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Field Name | Data type | Description |
| 1 | Name | VARCHAR (20) | Store the name. |
| 2 | Phone | VARCHAR (10) | Store the phone number. |
| 3 | Address | VARCHAR (20) | Store the address. |

**Book table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Field Name | Data type | Description |
| 1 | Book\_id | VARCHAR (10) | Store the book id. |
| 2 | Title | VARCHAR (20) | Store the book title. |
| 3 | Pub\_year | VARCHAR (5) | Store the publication year. |
| 4 | Publisher\_name | VARCHAR (20) | Store the publisher name. |

**Library\_branch table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Field Name | Data type | Description |
| 1 | Branch\_id | VARCHAR (10) | Store the branch id. |
| 2 | Branch\_name | VARCHAR (20) | Store the branch name. |
| 3 | Address | VARCHAR (20) | Store the branch address. |

**Card table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Field Name | Data type | Description |
| 1 | Card\_no | VARCHAR (10) | Store the card number. |
| 2 | Name | VARCHAR (20) | Store the name. |
| 3 | Age | VARCHAR (3) | Store the age. |
| 4 | Phone | VARCHAR (10) | Store the phone number. |
| 5 | Address | VARCHAR (20) | Store the address. |
| 6 | Occupation | VARCHAR (20) | Store the occupation. |
| 7 | Gender | VARCHAR (10) | Store the gender. |

**Issue table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Field name | Data type | Description |
| 1 | Book\_id | VARCHAR (10) | Store the book id. |
| 2 | Branch\_id | VARCHAR (10) | Store the branch id. |
| 3 | Card\_no | VARCHAR (10) | Store the card number. |
| 4 | Date\_out | VARCHAR (20) | Store the date of issue. |
| 5 | Due\_date | VARCHAR (20) | Store the date of return. |
| 6 | Return\_book | VARCHAR (5) | Store the status of book. |

**Chapter-6**

**SOURCE CODES**

**6.1 IMPORTING PACKAGES:**

import java.sql.\*;

import Project.ConnectionProvider;

import javax.swing.JOptionPane;

import java.text.SimpleDateFormat;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import javax.swing.JFrame;

import net.proteanit.sql.DbUtils;

**6.2 INSERT STATEMENTS:**

String card\_no = jTextField1.getText();

String name = jTextField2.getText();

String age = jTextField3.getText();

String phone = jTextField4.getText();

String address = jTextField5.getText();

String occupation = (String)jComboBox1.getSelectedItem();

String gender = (String)jComboBox2.getSelectedItem();

try

{

Connection con = ConnectionProvider.getCon();

Statement st = con.createStatement();

st.executeUpdate("insert into card values('"+card\_no+"','"+name+"','"+age+"','"+phone+"','"+address+"','"+occupation+"','"+gender+"')");

JOptionPane.showMessageDialog(null,"Successfully Updated");

setVisible(false);

new new\_card().setVisible(true);

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,"Card No. already exists");

setVisible(false);

new new\_card().setVisible(true);

}

**6.3 DELETE STATEMENTS:**

Int a=JOptionPane.showConfirmDialog(null,"Do you really want to delete?","Select",JOptionPane.YES\_NO\_OPTION);

if(a==0)

{

String card\_no=jTextField1.getText();

try

{

Connection con=ConnectionProvider.getCon();

Statement st=con.createStatement();

st.executeUpdate("delete from card where card\_no='"+card\_no+"'");

JOptionPane.showMessageDialog(null,"Successfully deleted");

setVisible(false);

new update\_delete\_card().setVisible(true);

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

}

**6.4 UPDATE STATEMENTS:**

String card\_no=jTextField1.getText();

String name=jTextField2.getText();

String age=jTextField3.getText();

String phone=jTextField4.getText();

String address=jTextField5.getText();

String occupation=(String)jComboBox1.getSelectedItem();

try

{

Connection con=ConnectionProvider.getCon();

PreparedStatement ps=con.prepareStatement("Update card set name=?,age=?,phone=?,address=?,occupation=? where card\_no=?");

ps.setString(1,name);

ps.setString(2,age);

ps.setString(3,phone);

ps.setString(4,address);

ps.setString(5,occupation);

ps.setString(6,card\_no);

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Successfully Updated");

setVisible(false);

new update\_delete\_card().setVisible(true);

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

**6.5 DISPLAY STATEMENTS:**

initComponents();

DefaultTableModel model=(DefaultTableModel)jTable1.getModel();

try

{

Connection con=ConnectionProvider.getCon();

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from card");

while(rs.next())

{

model.addRow(new Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7)});

}

}

catch(Exception e)

{

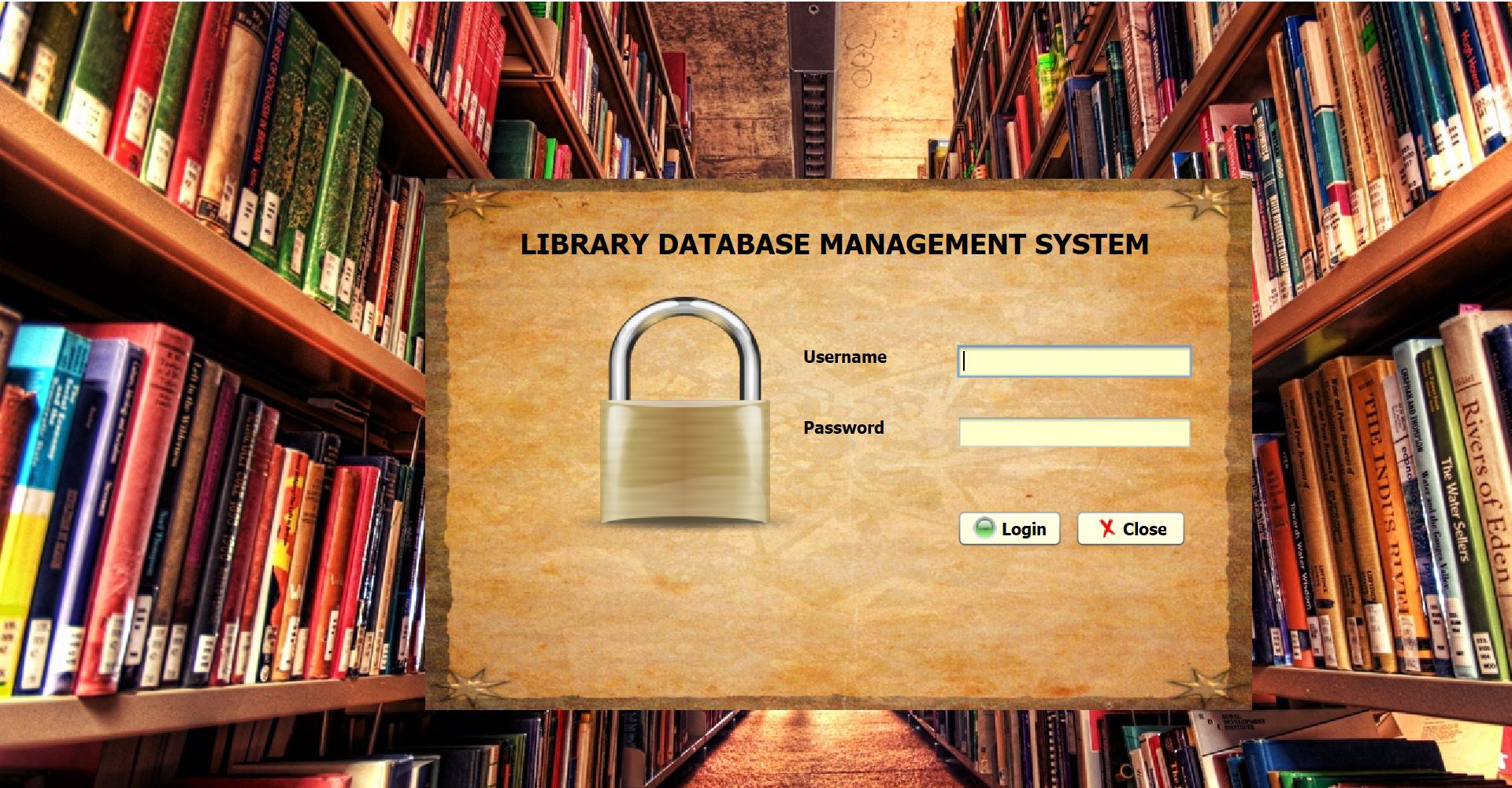
JOptionPane.showMessageDialog(null,e);

}

**Chapter-7**

**SNAPSHOTS**

**Admin Login Page: -**

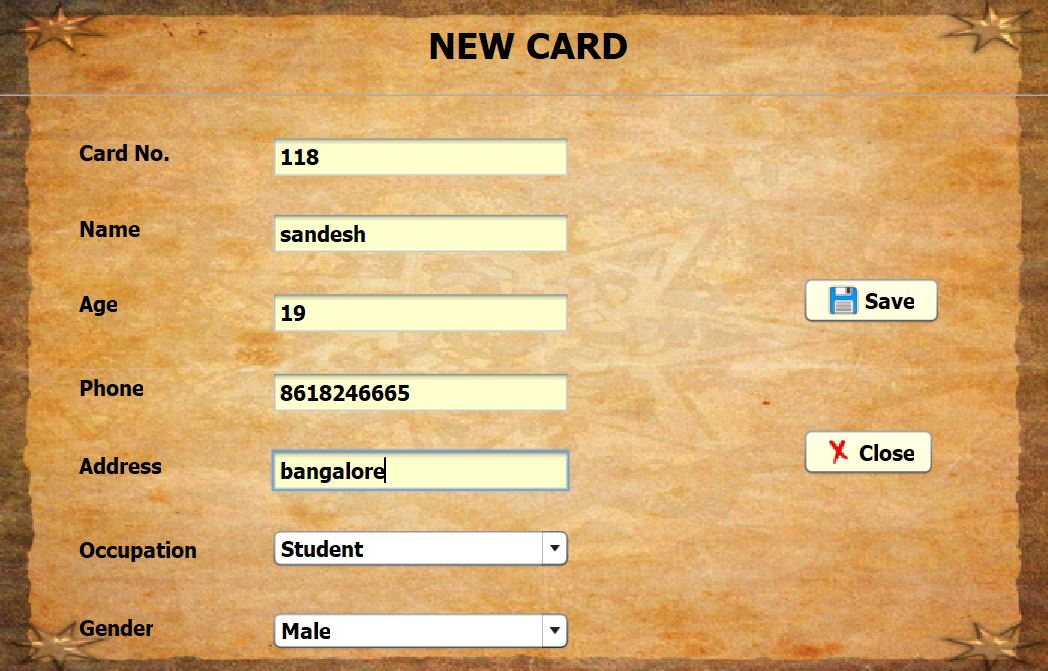


Admin need to enter user name and password and press login. If it is correct, then admin will be switched on to next page, if incorrect password then he will not be able to log in.

**Home Page: -**

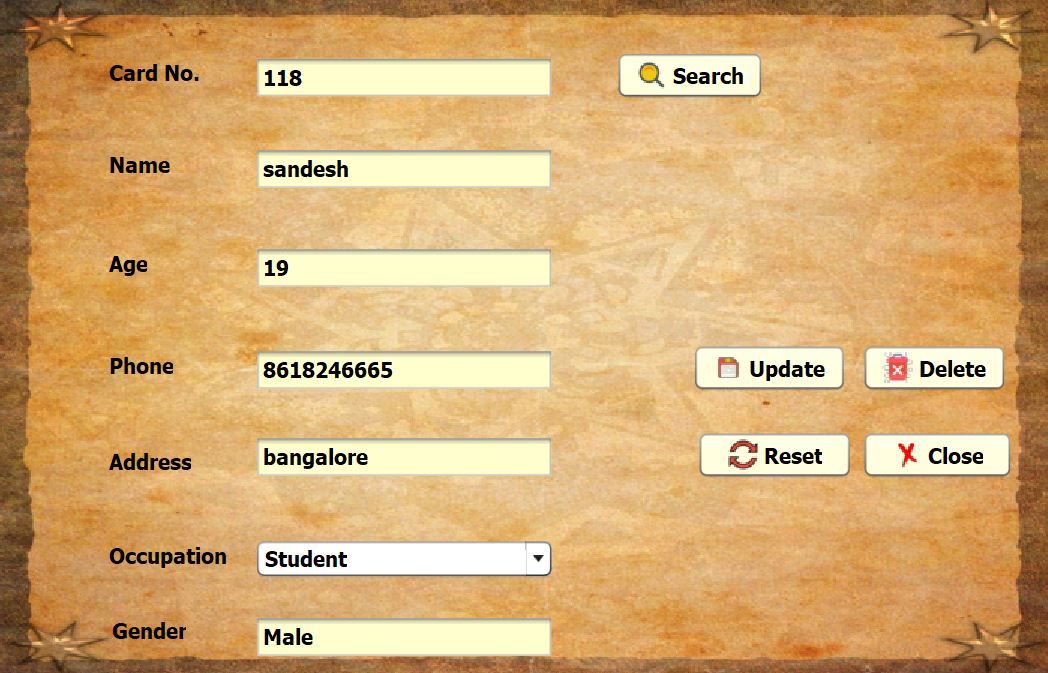
****This page shows all the entities of Library and details that should be mandatorily filled. The operations included are insert, delete, update and display. The admin can even logout of the session by pressing the log out icon.

**New entry Page: -**



New card can be added in the library, using this page. After entering details and pressing save, all the details will be saved in Library database.

**Update/Delete Page: -**

****

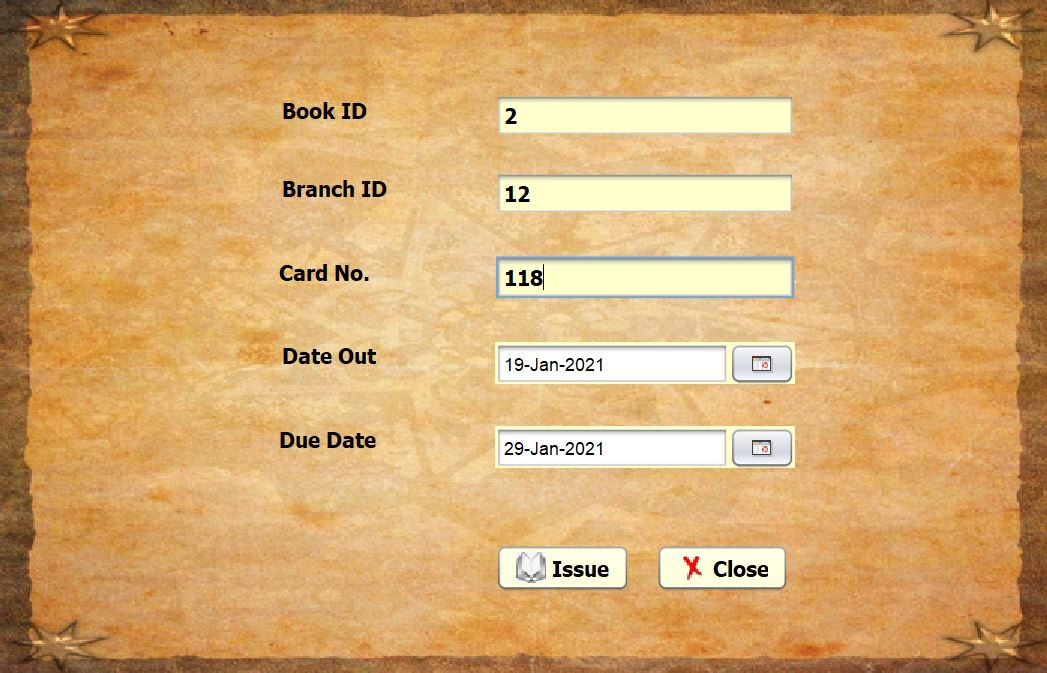
The details can be searched by using the primary key such as, card number and pressing search. All the details present in the database will be displayed in other fields, if the card number is valid. You can Update the details of that card number and can also delete the card number with all its details from the database.

**Display Page: -**



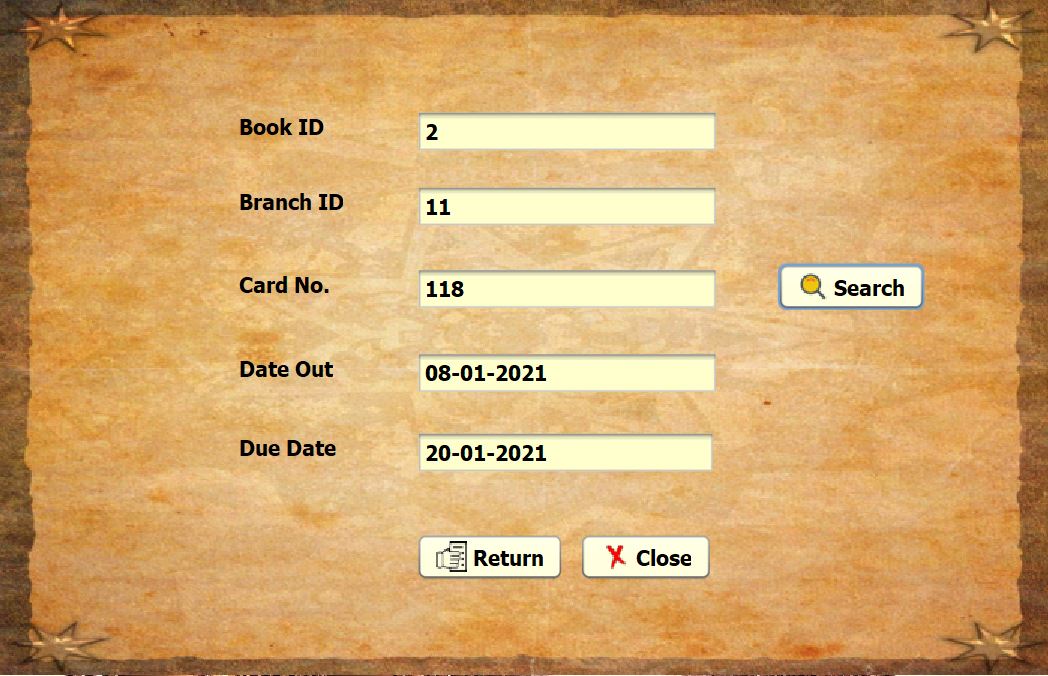
This page shows all the details of all the cards present in the Library database.

**Issue Page: -**

****

You can issue book to any card by entering the date out and due date details and press issue.

**Return Page: -**

****

Books can be returned back to the respective Library by using this page.

**Statistics Page: -**

****

All the details of books issued and returned will be displayed in this page.

**Chapter-8**

**CONCLUSION**

The Project “**Library Management System**” is designed in order to reduce the burden of maintaining bulk of records of all the Library management details in which Inserting, Retrieving and updating the details are easy when it is compared to the manual update and storing. This mini project helps in maintaining the book details as well as publishers and library branches and card details and also issued book details in an organized manner and to replace old paper work system.

During this semester, I have designed and implemented a database for managing a Library system, the ability to store books, and separate functions from customers and librarians. Furthermore, I was able to keep the system’s logic abstracted from both the end users and the DBMS by using MySQL connector to create a third-tier architecture. Finally, I used Java to design an applet-based GUI for remotely accessing the database.

The database design supports more operations than are currently implemented by the GUI. For example, the “hold” relationship can store a waiting list of customers who want to check out a particular media item; there is also a possibility of a customer having multiple library cards. If I were to continue with this project, I would implement the above features, and also improve the account management and search features.

**Chapter-9**

**REFERENCES**

* **Reference books: -**
* Java: The Complete reference
* Fundamentals of Database Management System
* **Websites: -**
* <https://stackoverflow.com/>
* <https://geeksforgeeks.com/>
* <https://quora.co.in>
* <https://github.com>
* <https://netbeans.org/>
* <https://www.mysql.com/>
* <http://oracle.com/>