**LIBRARY MANAGEMENT SYSTEM**

**MINI PROJECT REPORT**

Submitted by

**SHYAM SUNDAR S**

**17BCT046**

Under the Guidance of

**Dr. Mahalakshmi J** M.Sc., M.Phil., Ph.D.,

Assistant Professor

Department of Computer Technology

In partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF SCIENCE IN COMPUTER TECHNOLOGY**

Of Bharathiar University





**DEPARTMENT OF COMPUTER TECHNOLOGY**

**PSG COLLEGE OF ARTS & SCIENCE**

An Autonomous College-Affiliated to Bharathiar University

Accredited with ‘A’ grade by NAAC (3rd Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Coimbatore -641 014

**OCTOBER 2019**

**DEPARTMENT OF COMPUTER TECHNOLOGY**

**PSG COLLEGE OF ARTS & SCIENCE**

An Autonomous College-Affiliated to Bharathiar University

Accredited with A grade by NAAC (3rd Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Civil Aerodrome Post

Coimbatore -641 014

**CERTIFICATE**

This is to certify that this Project work entitled **“LIBRARY MANAGEMENT SYSTEM”** is a bonafide record of work done by **SHYAM SUNDAR S (17BCT046)** in partial fulfillment of the requirements for the award of Degree of **Bachelor of Science in Computer Technology** of Bharathiar University.

Faculty Guide Head of the Department

###### Submitted for Viva-Voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner External Examiner

**DECLARATION**

**DECLARATION**

I, **SHYAM SUNDAR S (17BCT046)** hereby declare that this Project work entitled **“LIBRARY MANAGEMENT SYSTEM”**, is submitted to PSG College of Arts & Science (Autonomous), Coimbatore in partial fulfillment for the award of Bachelor of Science in Computer Technology, is a record of original work done by me under the supervision and guidance of **Dr.** **Mahalakshmi J** M.Sc., M.Phil., Ph.D., Assistant Professor in Department of Computer Technology, PSG College of Arts & Science, Coimbatore.

This Project work has not been submitted by me for the award of any other Degree/ Diploma/ Associate ship/ Fellowship or any other similar degree to any other university.

PLACE : Coimbatore **SHYAM SUNDAR S**

DATE : **17BCT046**

**ACKNOWLEDGEMENT**

**ACKNOWLEDGEMENT**

My venture stands imperfect without dedicating my gratitude to a few people who have contributed a lot towards the victorious completion for my project work.

I would like to thank **Mr. L. Gopalakrishnan**, **Managing Trustee, PSG & Sons Charities**, for providing me a prospect and surroundings that made the work possible.

I take this opportunity to express my deep sense of gratitude to **Dr. T. Kannaian,** Secretary of PSG College of Arts and Science, Coimbatore for permitting and doing the needful towards the successful completion of this project.

I express my deep sense of gratitude and sincere thanks to our Principal **Dr. D. Brindha MSc, MPhil, PhD,** for her valuable advice and concern on students.

I am very thankful to **Dr. A Anguraj, MSc, MPhil, PhD, Vice Principal**, **Prof. M Umarani, MCom, MPhil, Faculty-In-Charge (Student Affairs)** for their support.

I kindly and sincerely thank **Dr. S. Geetharani MCA, MPhil, PhD, Head of the**

**Department of Computer Technology** for her whole hearted help to complete this project successfully by giving valuable suggestions.

I convey my heartiest and passionate sense of thankfulness to my project guide **Dr. Mahalakshmi J** M.Sc., M.Phil., Ph.D., Assistant Professor, Department of Computer Technology, for her timely suggestion which had enable me in completing the project successfully.

This note of acknowledgement will be incomplete without paying my heartful devotion to my parents, my friends and other people, for their blessings, encouragement, financial support and the patience, without which it would have been impossible for me to complete the job.

**SYNOPSIS**

**SYNOPSIS**

The main objective of this “LIBRARY MANAGEMENT SYSTEM” project is to develop the Software for the library which requires the information management of the books and members in the library. This project work aim to design and implement a computerized library management system.

It has a facility of admin login through which the admin can monitor the whole system. In this computerized system where we can record various transactions like issue of books, return books, addition of members. It also has facility of an online notice board where members are not renew or submission of book.

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.

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.

Overall this project of ours is being developed to help members and librarians to maintain the library in the best way possible and also reduce the human works.

**TABLE OF CONTENTS**

**TABLE OF CONTENTS**

**CONTENTS**  **PAGE NO.**

1. **INTRODUCTION**

1.1. Project Overview 1

1.2. Module Description 2

1. **SYSTEM SPECIFICATION**
   1. Hardware Specification 4
   2. Software Specification 4
   3. Software Description 5
2. **SYSTEM ANALYSIS**
   1. Existing System 7
   2. Proposed System 7
3. **SYSTEM DESIGN** 
   1. Data Flow Diagram 9
   2. ER Diagram 11
   3. Table Design 13
   4. Form Design 14
4. **SYSTEM TESTING & IMPLEMENTATION**  17
5. **SCOPE FOR FUTURE ENHANCEMENTS**  21
6. **CONCLUSION**  23

**BIBLIOGRAPHY** 25

**APPENDICES** 27

A. SCREEN SHOTS 27

B. SAMPLE CODING 31

**INTRODUCTION**

**1. INTRODUCTION**

* 1. **PROJECT OVERVIEW**

The aim of the project is to create a user friendly and efficient application. This is the application which managing library books and members. In this application admin can add books and library members and can issue the books. Books and user maintenance modules are also included in this system which would keep track of the users 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. Admin can send the overdue notification to the user which is already get from member and stored in member table.

**1.2. MODULE DESCRIPTION**

**Add Book**

This module can add books with details like book id, book name, book author and publisher. And stores in book database table.

**Add Member**

This module will add members to our library and will stored in member table with member name, member id, mobile and email.

**View Book**

This module helps to view the books stored in the book database table.

**View Members**

We can view members of the library which stored in the member database table.

**Book Issue**

Through this module admin can issue the book which is available in library and issue with book id and member id and can store the data in issue table.

**View Issued Books**

This can view the issue book list retrieve from issue table.

**Renew / Submission**

This module can renew or submission of book which is already issued.

**Overdue notification**

This module is to notify or alert the members in library with email to renew or submit the book.

**SYSTEM CONFIGURATION**

**2. SYSTEM CONFIGURATION**

**2.1 HARDWARE CONFIGURATION**

**Processor** **:** Intel i3

**CPU Clock Speed** **:** 1.08GHz

**RAM** **:** 1 GB

**Hard Disk Drive**  **:** 500MB

**Hardware**  **:** Mouse & Keyboard

**2.2 SOFTWARE CONFIGURATION**

**Operating System**  **:** Windows 7 and above

**Front End** **:** JavaFx

**Tool :** Netbeans IDE, Scene Builder

**2.3 SOFTWARE DESCRIPTION**

**Front End**

**JavaFX**

JavaFX is a Java library used to develop Desktop applications as well as Rich Internet Applications (RIA). The applications built in JavaFX, can run on multiple platforms including Web, Mobile and Desktops.

  JavaFX library such as Fundamentals, 2D Shapes, 3D Shapes, Effects, Animation, Text, Layouts, UI Controls, Transformations, Charts, JavaFX with CSS, JavaFX with Media etc.

JavaFX is intended to replace swing in Java applications as a GUI framework. However, It provides more functionalities than swing. Like Swing, JavaFX also provides its own components and doesn't depend upon the operating system. It is lightweight and hardware accelerated. It supports various operating systems including Windows, Linux and Mac OS.

FXML is the XML based Declarative markup language. The coding can be done in FXML to provide the more enhanced GUI to the user

Scene Builder generates FXML mark-up which can be ported to an IDE.

**Back End**

**Derby Database**

An embedded database, on the other hand, is incorporated into the software program by the developer in such a way that it is supposed to be invisible to the application end user. Being oblivious of the underlying database, there is no need of any external interface to maintain the database separately. So here, our only interest is the database engine; if that can be embedded and run with the application, we get what we want—an embedded database application.

**SYSTEM ANALYSIS**

**3. SYSTEM ANALYSIS**

**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. Here the key question is- what all problems exist in the present system what must be done to solve the problem Analysis begins when a user or manager begins a study of the program using existing system. 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.

**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. Our proposed system has the following advantages.

• User friendly interface

• Fast access to database

• Less error

• More Storage Capacity

• Search facility

• Look and Feel Environment

All the manual difficulties in managing the Library have been rectified by implementing computerization.

**SYSTEM DESIGN**

**4. SYSTEM DESIGN**

**4.1 DATA FLOW DIAGRAM**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the [visualization](https://en.wikipedia.org/wiki/Data_visualization) of [data processing](https://en.wikipedia.org/wiki/Data_processing).

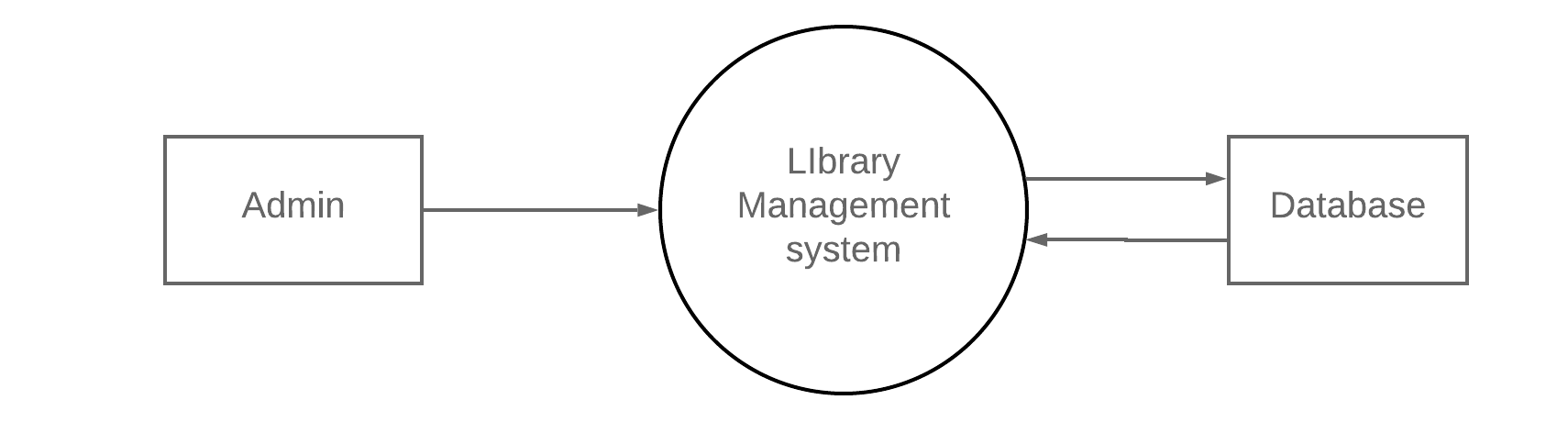
**-** Entity (A source of data or destination of data)

**-** Process (A process or task that is performed by the system)

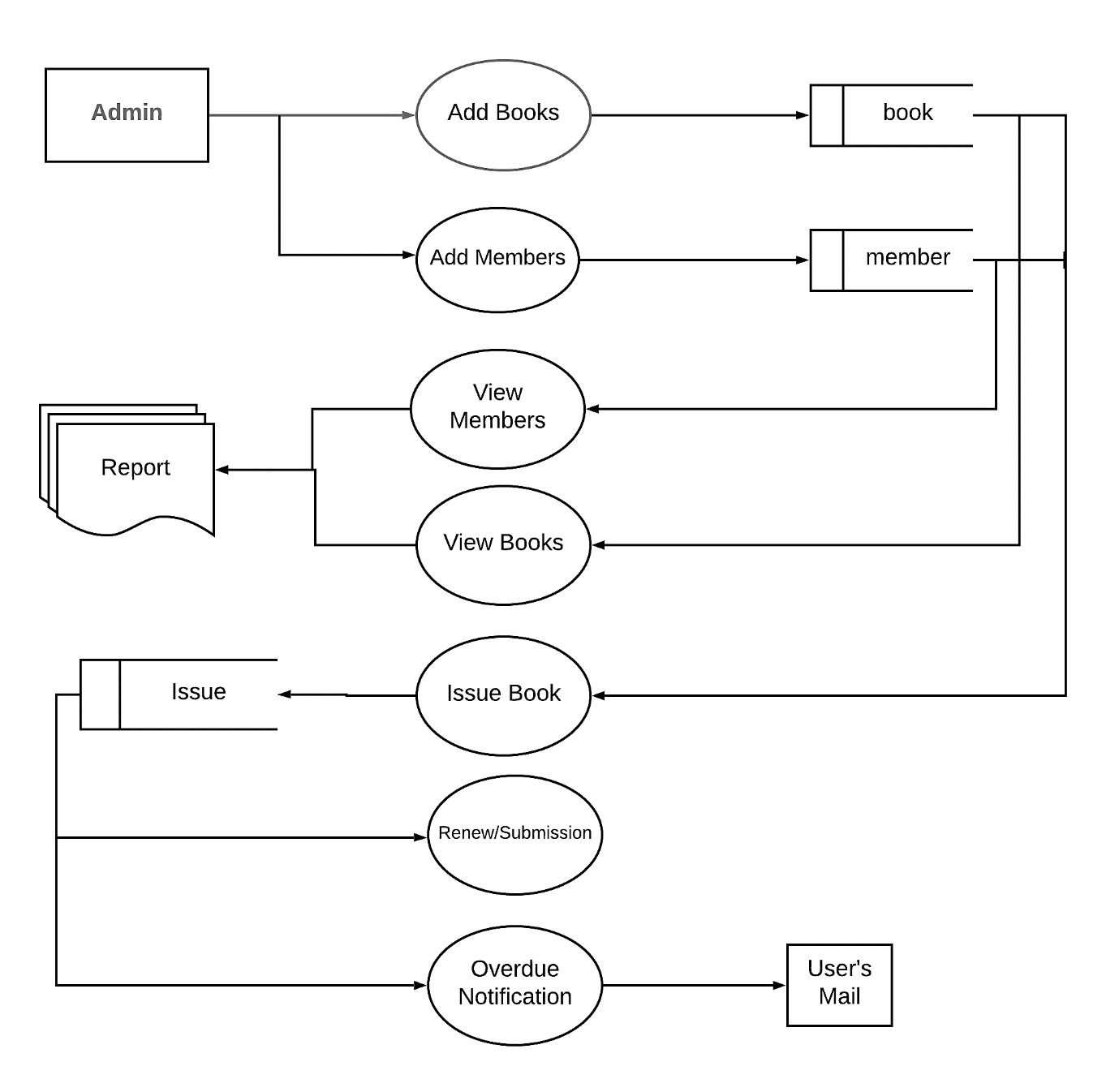
- Data Store (Data at rest being stored for later use)

**-** Data Flow (Data flowing from place to place such as input or

Output to a process)



Data Flow Diagram – Level 0



Data Flow Diagram – Level 1

**4.2 ER DIAGRAM**

ER- Diagram is a visual representation of data that describe how data is related to each other.

* **Rectangles:** This symbol represent entity types
* **Ellipses :** Symbolrepresent attributes
* **Diamonds:** This symbolrepresents relationship types
* **Lines:** It links attributes to entity types and entity types with other relationship types
* **Primary key:** attributes are underlined
* **Double Ellipses:** Represent multi-valued attributes



RENEW

Yes

M

M

M

M

M

M

1

No

Notification

Book\_renewal

Book issue

ISSUE

BOOKS

MEMBER

MANAGE

ADMIN

**4.3 TABLE DESIGN**

**Table name:** book\_list

**Description:** This table stores the data about book such as book title, book id, author, and publisher.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Attributes | Description |
| title | varchar(50) | Not null | Book title |
| book\_id | varchar(10) | Primary key | Book id |
| author | varchar(50) | Not null | Author name |
| publisher | varchar(50) | Not null | publisher |

**Table name:** member\_list

**Description:** This table stores the data about member details like member name, member id, mobile and email.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Attributes | Description |
| name | varchar(50) | Not null | Member name |
| member\_id | varchar(10) | Primary key | Member id |
| mobile | varchar(20) | Not null | mobile |
| email | varchar(50) | Not null | email |

**Table name:** issued\_books

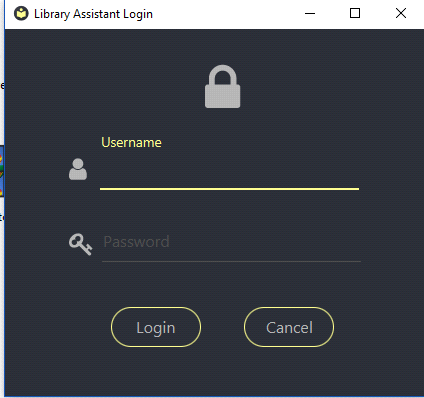
**Description:** This table stores the data about book issues which is book id, book title, and whom to issued, date of issue, days

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Attributes | Description |
| book\_id | varchar(50) | Primary key | Book id |
| title | varchar(10) | Not null | Book title |
| issued\_to | varchar(20) | Not null | Member Id |
| doi | date | Not null | Date of issue |
| days | number | Not null | Fine days |

**4.4 FORM DESIGN**

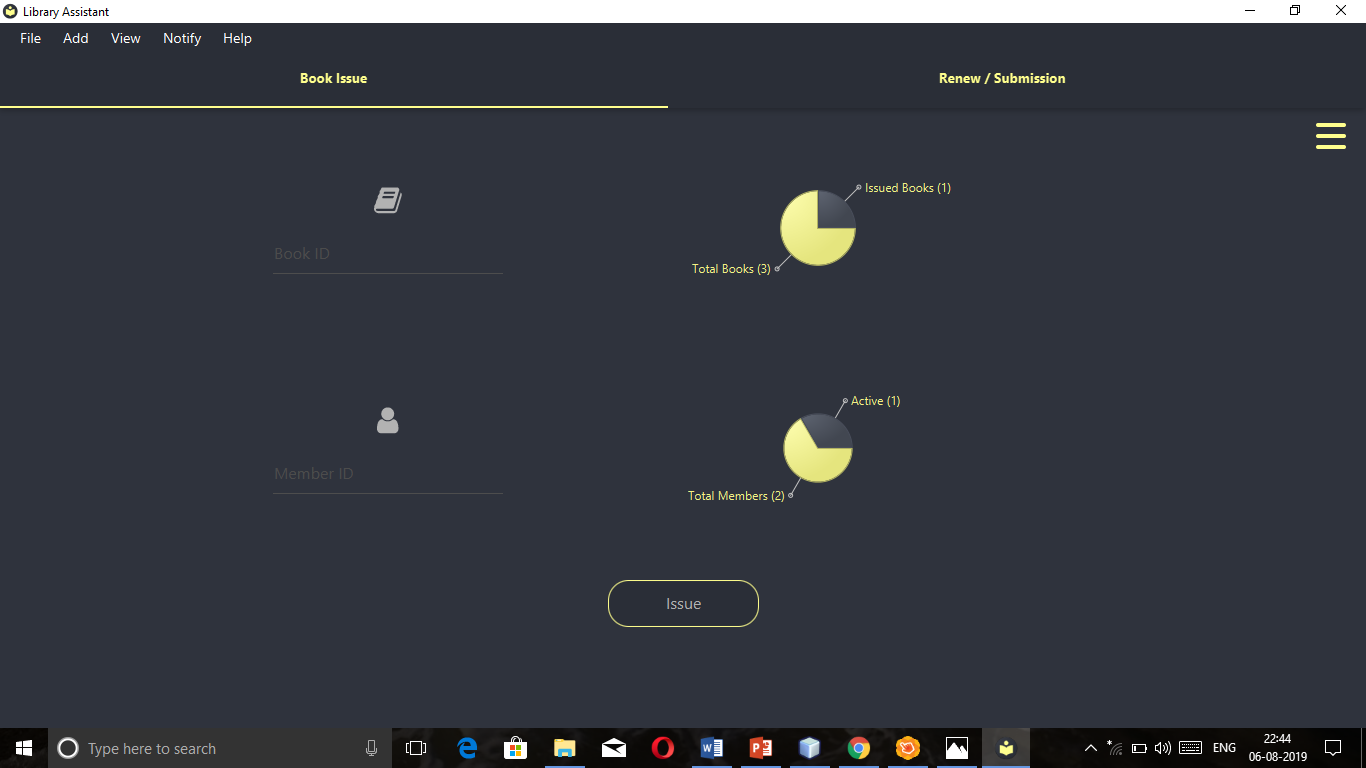
**Login form**

This is the login form, only admin can logon to the system with credentials.

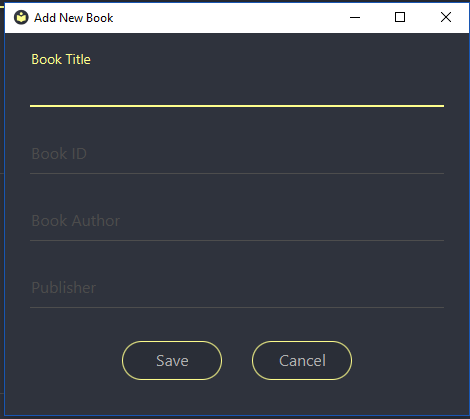


**Main form**

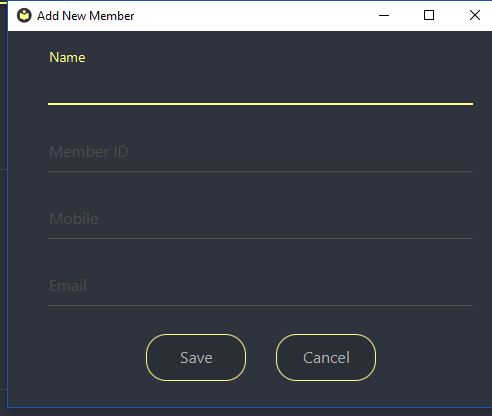
This is the main module, you can choose functions in our computerized system.



**Add book form:** In this form librarian can add books to the system.



**Add member form:** In this form librarian can add members in the library to the system.



**SYSTEM TESTING**

**5. SYSTEM TESTING AND IMPLEMENTATION**

The system testing verifies the whole set of programs that hang together. Before the system is acceptable by the user, testing is very important, it eliminates communicational problem, programmer’s negligence or time constraints, which causes error. The strategies for testing include unit testing, integration testing, system testing, implementation testing.

**SYSTEM TESTING**

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works, According to the actual requirement and objectives of the system.

The philosophy behind testing is to find the errors. A good test is one that undiscovered error. Test cases are devised with this purpose in mind. Test cases are is a set of data that the system will process as an input. However the data are created with the intent of determining whether the system will process them correctly without any errors to produce the required output.

Testing could be viewed as destructive rather than constructive. It is the process of executing a programmed with the intend of finding errors. The testing is one that will uncover different classes of errors with minimum amount of time and effort. In the proposed system testing is done. Testing is performed to ensure that software function appear to be working according to the specifications and that the performance requirement of the system.

**TESTING METHODOLOGIES**

**Black Box Testing**

Black box testing also called behavioral testing focuses on the functional requirements of the software. That is black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing attempts to find errors in the following categories. Incorrect or missing functions. Interface errors. Errors in data structures or external data base access Behavior or performance errors. Initialization and termination errors.

Functional Testing and black box type testing geared to functional requirements of an application. This type of testing should be done by testers. Our project does the functional testing of what input given and what output should be obtained.

System Testing-black box type testing that is based on overall requirements specifications; covers all combined parts of a system. The system testing to be done here is that to check with all the peripherals used in the project.

**WHITE BOX TESTING**

White box testing sometimes called glass box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that guarantee that all independent paths within a module have been exercised at least once. Exercise all logical decisions on their true and false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structures to ensure their validity.

**UNIT TESTING**

The most „micro‟ scale of testing to test particular functions or code modules. Typically, it is done by the programmer and not by tester, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code; may require developing test modules or test harnesses.

**QUALITY ASSURANCE**

Software Quality Assurance involves the entire software development process monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and deal with. It is oriented to „prevention‟.

**SOFTWARE LIFE CYCLE**

The life cycle begins when an application is first conceived and ends when it is no longer in use. It includes aspects such as initial concept, requirements analysis, functional design, internal design, documentation planning, test planning, coding, document preparation, integration, testing, maintenance, updates, retesting, phase-out, and other aspects.

**VERIFICATION AND VALIDATION**

Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensures that the software has been built is traceable to customer requirements.

Verification and validation encompasses a wide array of SQA activities that include formal technical reviews, quality and configuration audits, performance monitoring, simulation, feasibility study, documentation review, database review, algorithm analysis, development testing, qualification testing and installation testing.

**SYSTEM IMPLEMENTATION**

System implementation is a stage in a stage in the project where the where the theoretical designs turned into working system. The most crucial stage the user confidence that the new system will work effectively and efficiently.

The performance of reliability of the system was tested and it gained acceptance. The system was implemented successfully. Implementation is a process that means converting a new system into operation.

Proper implementation is essential to provide a reliable system to meet organization requirements. During the implementation stage a live demon was undertaken and made in front of end-users.

**IMPLEMENTATION**

Implementation is the stage in the paper where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively

**SCOPE FOR FUTURE ENHANCEMENT**

**6. SCOPE FOR FUTURE ENHANCEMENTS**

In future it is a possible one to add new features without any problem with enhanced. As the technology used is a good one it is flexible for future enhancement and it is also possible to alter the front-end and back-end without any problem.

This project currently used for scanning viruses. In future we can add some additional features also based on the user requirements.

**CONCLUSION**

1. **CONCLUSION**

Library Management System with the functional modules was successfully developed as a secured, digitalized and user friendly system for the governing of public library of Jaffna. Manually maintained operations were computerized. Thus the system created to overcome the problems effectively without any corrupted data or information.

**BIBLIOGRAPHY**

**BIBILOGRAPHY**

The following books were referred during the analysis and execution phase of the project.

**REFERENCE:**

* **JavaFX For Dummies** By Doug Lowe 2014 edition
* **JavaFX Essentials** By Mohamed Taman year-2015

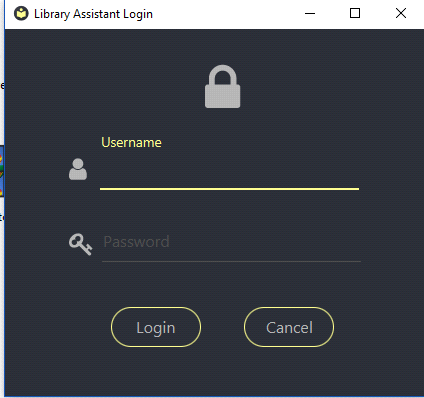
**WEBSITES:**

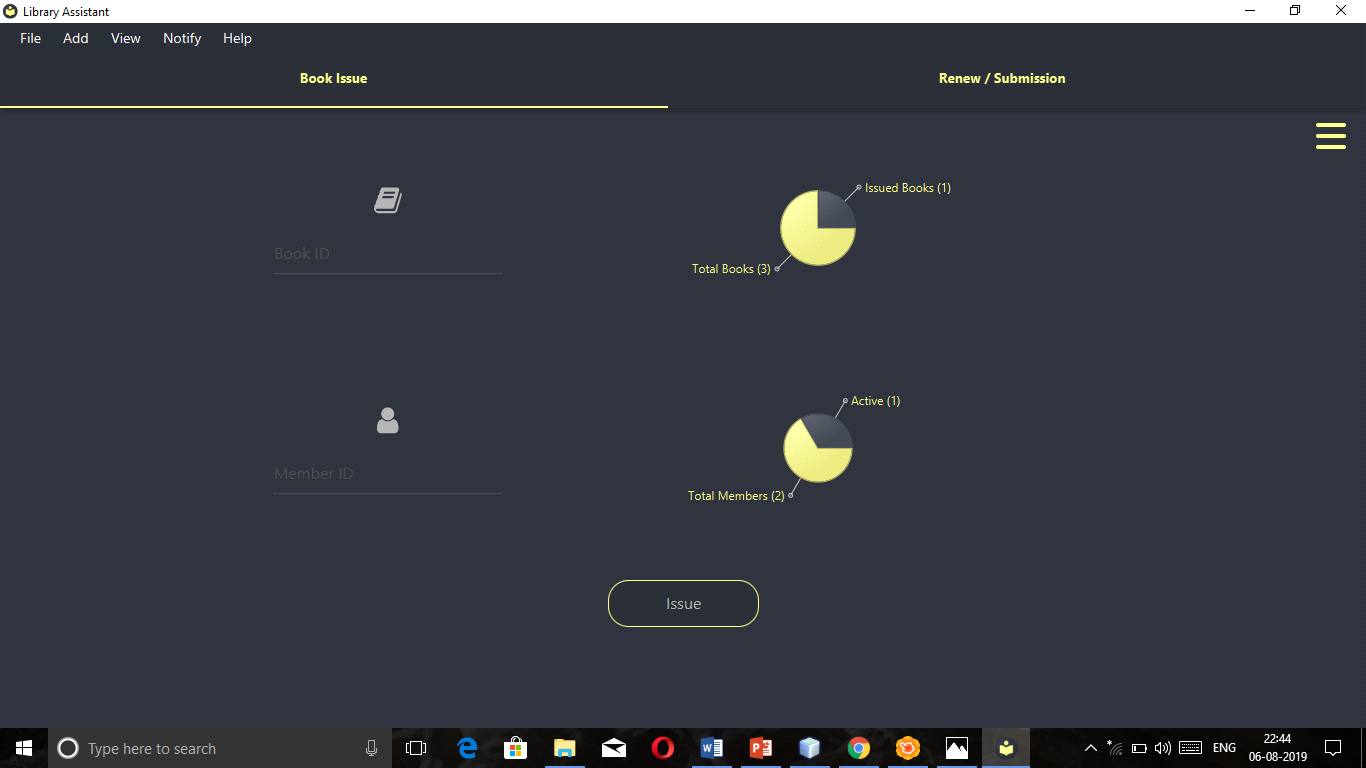
* <http://www.tutorialspoint.com/javafx>
* <http://www.udemy.com>
* [http://www.javatpoint.com](http://www.javatpoint.com/)
* <http://www.genuinecoder.com>
* <http://www.github.com/afsalashyana>

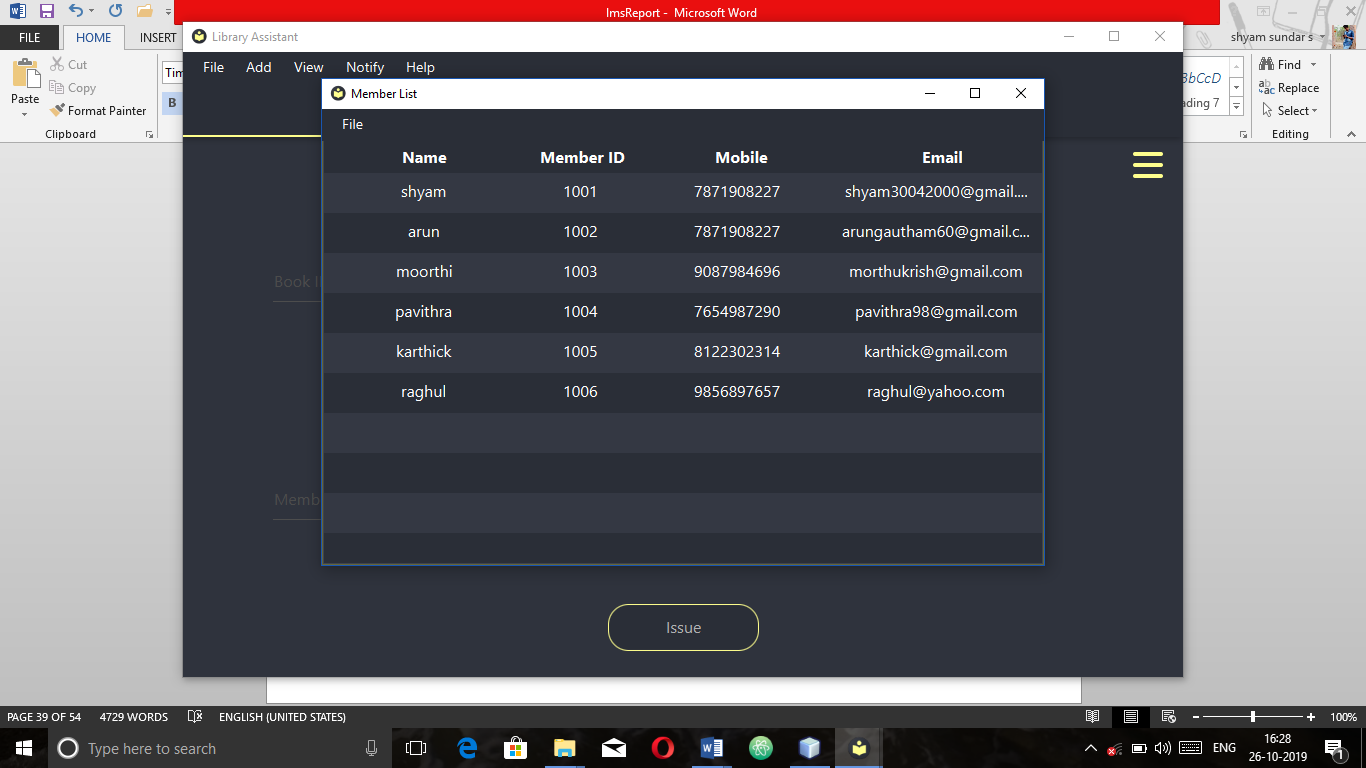
**APPENDICES**

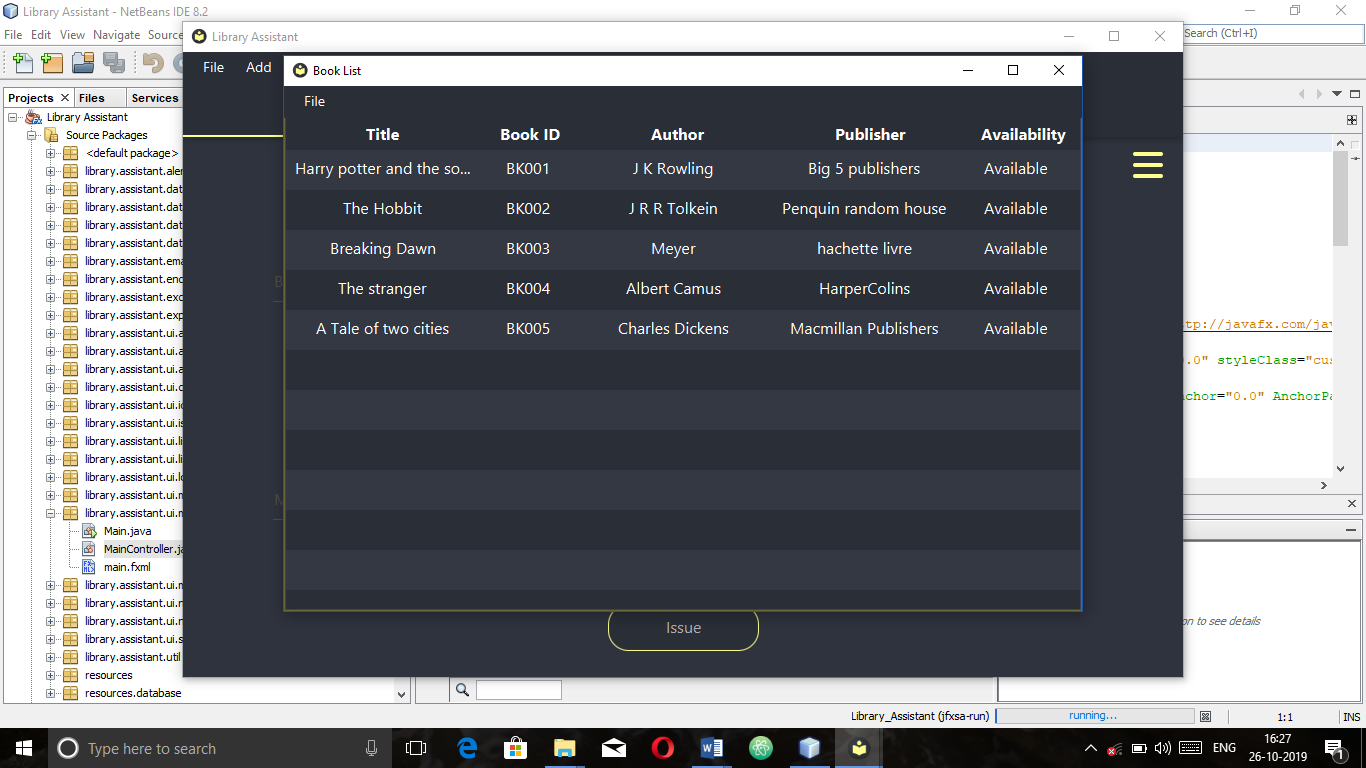
**APPENDICES:**

**SCREENSHOTS**

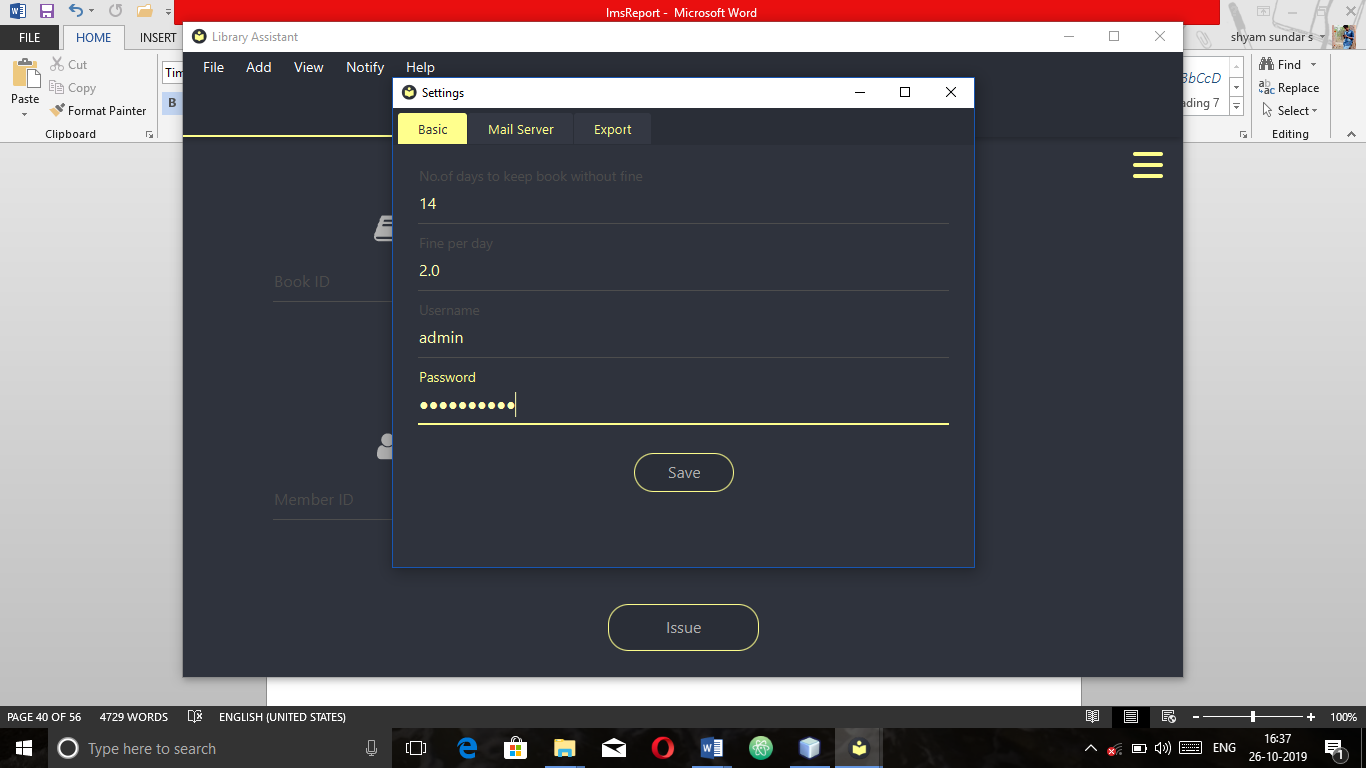


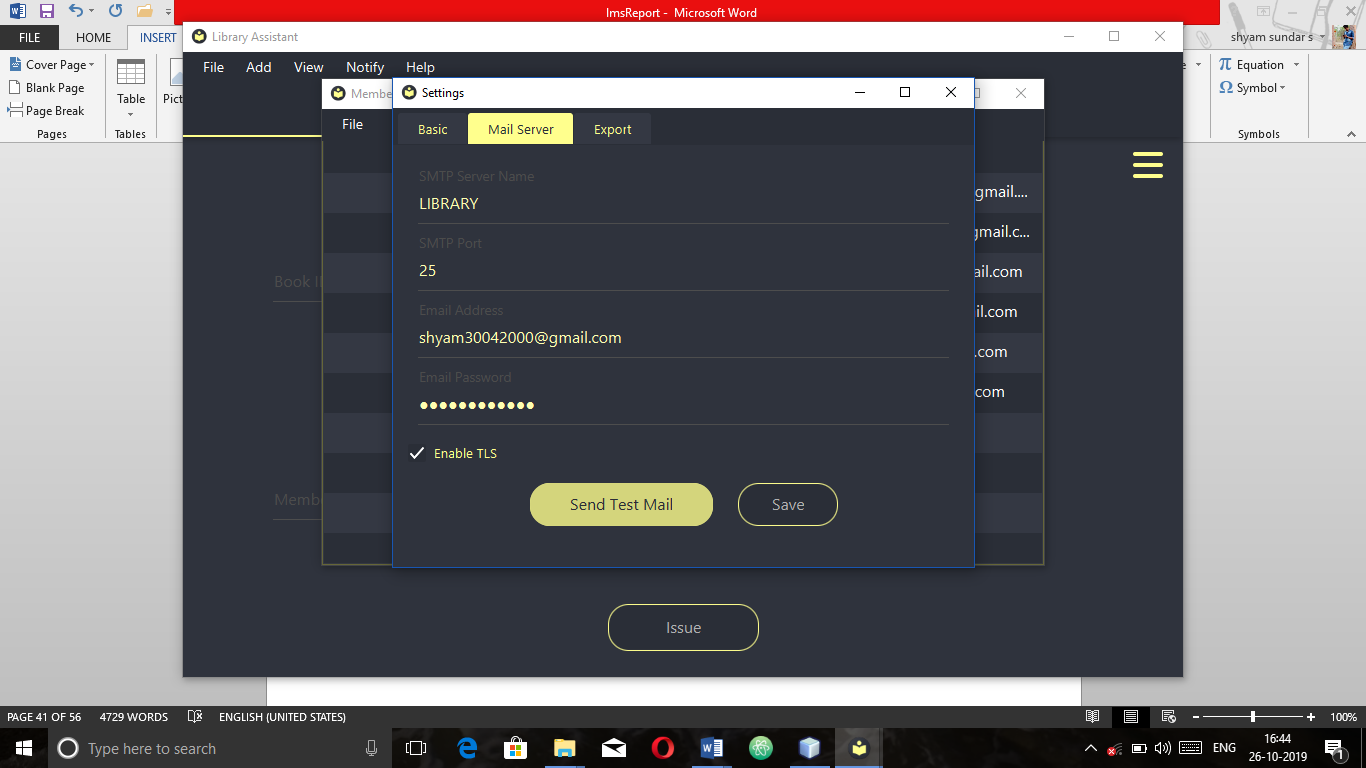












**SAMPLE CODING**

**Main interface**

<?xml version="1.0" encoding="UTF-8"?>

<?import com.jfoenix.controls.JFXButton?>

<?import com.jfoenix.controls.JFXDrawer?>

<?import com.jfoenix.controls.JFXHamburger?>

<?import com.jfoenix.controls.JFXTabPane?>

<?import com.jfoenix.controls.JFXTextField?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIconView?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Menu?>

<?import javafx.scene.control.MenuBar?>

<?import javafx.scene.control.MenuItem?>

<?import javafx.scene.control.Tab?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.layout.BorderPane?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.StackPane?>

<?import javafx.scene.layout.VBox?>

<?import javafx.scene.text.Text?>

<StackPane fx:id="rootPane" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="625.0" prefWidth="1000.0" styleClass="custom-pane" stylesheets="@../../../../resources/dark-theme.css" xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1" fx:controller="library.assistant.ui.main.MainController">

<children>

<JFXDrawer fx:id="drawer" defaultDrawerSize="150.0" resizableOnDrag="true" resizeContent="true" styleClass="jfx-drawer" />

<AnchorPane fx:id="rootAnchorPane" prefHeight="200.0" prefWidth="200.0">

<children>

<AnchorPane AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="25.0">

<children>

<JFXTabPane fx:id="mainTabPane" AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">

<tabs>

<Tab fx:id="bookIssueTab" closable="false" styleClass="tab-main" text="Book Issue">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">

<children>

<VBox prefHeight="412.0" prefWidth="557.0" AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">

<children>

<HBox fx:id="book\_info" alignment="CENTER" prefHeight="200.0" prefWidth="1000.0">

<children>

<VBox alignment="CENTER" spacing="20.0">

<HBox.margin>

<Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />

</HBox.margin>

<children>

<FontAwesomeIconView glyphName="BOOK" size="30" />

<JFXTextField fx:id="bookIDInput" labelFloat="true" onAction="#loadBookInfo" prefHeight="26.0" prefWidth="230.0" promptText="Book ID" />

</children>

</VBox>

<StackPane fx:id="bookInfoContainer">

<HBox.margin>

<Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />

</HBox.margin>

<children>

<VBox alignment="CENTER" prefHeight="147.0" prefWidth="550.0">

<children>

<Text fx:id="bookName" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" textAlignment="CENTER" wrappingWidth="460.0">

<VBox.margin>

<Insets top="10.0" />

</VBox.margin>

</Text>

<Text fx:id="bookAuthor" layoutX="10.0" layoutY="71.0" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" textAlignment="CENTER" wrappingWidth="460.0">

<VBox.margin>

<Insets top="10.0" />

</VBox.margin>

</Text>

<Text fx:id="bookStatus" layoutX="10.0" layoutY="91.0" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" textAlignment="CENTER" wrappingWidth="460.0">

<VBox.margin>

<Insets top="10.0" />

</VBox.margin>

</Text>

</children>

</VBox>

</children>

</StackPane>

</children>

<VBox.margin>

<Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />

</VBox.margin>

</HBox>

<HBox fx:id="member\_info" alignment="CENTER" layoutX="10.0" layoutY="10.0" prefHeight="200.0" prefWidth="1000.0">

<children>

<VBox alignment="CENTER" spacing="20.0">

<HBox.margin>

<Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />

</HBox.margin>

<children>

<FontAwesomeIconView glyphName="USER" size="30" />

<JFXTextField fx:id="memberIDInput" labelFloat="true" onAction="#loadMemberInfo" prefWidth="230.0" promptText="Member ID" />

</children>

</VBox>

<StackPane fx:id="memberInfoContainer">

<HBox.margin>

<Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />

</HBox.margin>

<children>

<VBox alignment="CENTER" prefHeight="151.0" prefWidth="550.0">

<children>

<Text fx:id="memberName" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" textAlignment="CENTER" wrappingWidth="460.0">

<VBox.margin>

<Insets top="10.0" />

</VBox.margin>

</Text>

<Text fx:id="memberMobile" layoutX="10.0" layoutY="71.0" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" textAlignment="CENTER" wrappingWidth="460.0">

<VBox.margin>

<Insets top="10.0" />

</VBox.margin>

</Text>

</children>

</VBox>

</children>

</StackPane>

</children>

<VBox.margin>

<Insets bottom="5.0" left="20.0" right="20.0" />

</VBox.margin>

</HBox>

<HBox alignment="CENTER" prefHeight="100.0" prefWidth="200.0">

<children>

<JFXButton fx:id="btnIssue" onAction="#loadIssueOperation" onKeyPressed="#handleIssueButtonKeyPress" prefHeight="44.0" prefWidth="151.0" text="Issue">

<HBox.margin>

<Insets />

</HBox.margin>

<padding>

<Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />

</padding>

</JFXButton>

</children>

</HBox>

</children>

</VBox>

</children>

</AnchorPane>

</content>

</Tab>

<Tab fx:id="renewTab" closable="false" styleClass="tab-main" text="Renew / Submission">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">

<children>

<BorderPane layoutX="251.0" layoutY="71.0" prefHeight="412.0" prefWidth="557.0" AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">

<top>

<JFXTextField fx:id="bookID" labelFloat="true" maxWidth="274.0" minWidth="274.0" onAction="#loadBookInfo2" prefHeight="32.0" prefWidth="274.0" promptText="Enter BOOK ID" BorderPane.alignment="CENTER">

<BorderPane.margin>

<Insets top="30.0" />

</BorderPane.margin>

</JFXTextField>

</top>

<bottom>

<HBox alignment="CENTER" prefHeight="69.0" prefWidth="557.0" BorderPane.alignment="CENTER">

<children>

<JFXButton fx:id="renewButton" disable="true" onAction="#loadRenewOp" prefWidth="150.0" text="Renew">

<HBox.margin>

<Insets right="20.0" />

</HBox.margin>

</JFXButton>

<JFXButton fx:id="submissionButton" disable="true" onAction="#loadSubmissionOp" prefWidth="150.0" text="Submission">

<HBox.margin>

<Insets right="20.0" />

</HBox.margin>

</JFXButton>

</children>

</HBox>

</bottom>

<center>

<HBox fx:id="submissionDataContainer" alignment="CENTER" opacity="0.0" prefHeight="10000.0" prefWidth="10000.0" BorderPane.alignment="CENTER">

<children>

<VBox alignment="TOP\_CENTER" prefHeight="336.0" prefWidth="234.0" spacing="20.0" styleClass="submission-container">

<children>

<FontAwesomeIconView glyphName="USER" size="60">

<VBox.margin>

<Insets top="20.0" />

</VBox.margin>

</FontAwesomeIconView>

<Text fx:id="memberNameHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Member Name Holder" wrappingWidth="300.0" />

<Text fx:id="memberEmailHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Member Email Holder" wrappingWidth="300.0" />

<Text fx:id="memberContactHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Member Contact" wrappingWidth="300.0" />

</children>

<HBox.margin>

<Insets right="20.0" />

</HBox.margin>

</VBox>

<VBox alignment="TOP\_CENTER" prefHeight="336.0" prefWidth="248.0" spacing="20.0" styleClass="submission-container">

<children>

<FontAwesomeIconView glyphName="BOOK" size="60">

<VBox.margin>

<Insets top="20.0" />

</VBox.margin>

</FontAwesomeIconView>

<Text fx:id="bookNameHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Book Name Holder" wrappingWidth="300.0" />

<Text fx:id="bookAuthorHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Book Author Holder" wrappingWidth="300.0" />

<Text fx:id="bookPublisherHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Book Publisher Holder" wrappingWidth="300.0" />

</children>

<HBox.margin>

<Insets right="20.0" />

</HBox.margin>

</VBox>

<VBox alignment="TOP\_CENTER" prefHeight="336.0" prefWidth="248.0" spacing="20.0" styleClass="submission-container">

<children>

<FontAwesomeIconView glyphName="CALENDAR\_CHECK\_ALT" size="60">

<VBox.margin>

<Insets top="20.0" />

</VBox.margin>

</FontAwesomeIconView>

<Text fx:id="issueDateHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Issue Date Holder" wrappingWidth="300.0" />

<Text fx:id="numberDaysHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="No of Days Holder" wrappingWidth="300.0" />

<Text fx:id="fineInfoHolder" strokeType="OUTSIDE" strokeWidth="0.0" styleClass="main-text" text="Fine Holder" wrappingWidth="300.0" />

</children>

</VBox>

</children>

<BorderPane.margin>

<Insets top="20.0" />

</BorderPane.margin>

</HBox>

</center>

</BorderPane>

</children>

</AnchorPane>

</content>

</Tab>

</tabs>

</JFXTabPane>

<JFXHamburger fx:id="hamburger" styleClass="jfx-hamburger-icon" AnchorPane.rightAnchor="20.0" AnchorPane.topAnchor="75.0" />

</children>

</AnchorPane>

<MenuBar AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">

<menus>

<Menu mnemonicParsing="false" text="File">

<items>

<MenuItem mnemonicParsing="false" onAction="#handleMenuSettings" text="Settings" />

<MenuItem mnemonicParsing="false" onAction="#handleMenuClose" text="Exit" />

</items>

</Menu>

<Menu mnemonicParsing="false" text="Add">

<items>

<MenuItem mnemonicParsing="false" onAction="#handleMenuAddBook" text="Add Book" />

<MenuItem mnemonicParsing="false" onAction="#handleMenuAddMember" text="Add Member" />

</items>

</Menu>

<Menu mnemonicParsing="false" text="View">

<items>

<MenuItem mnemonicParsing="false" onAction="#handleMenuViewBook" text="Book List" />

<MenuItem mnemonicParsing="false" onAction="#handleMenuViewMemberList" text="Member List" />

<MenuItem mnemonicParsing="false" onAction="#handleIssuedList" text="Issued Book List" />

<MenuItem mnemonicParsing="false" onAction="#handleMenuFullScreen" text="Full Screen" />

</items>

</Menu>

<Menu mnemonicParsing="false" text="Notify">

<items>

<MenuItem mnemonicParsing="false" onAction="#handleMenuOverdueNotification" text="Overdue Notification" />

</items>

</Menu>

<Menu mnemonicParsing="false" text="Help">

<items>

<MenuItem mnemonicParsing="false" onAction="#handleAboutMenu" text="About" />

</items>

</Menu>

</menus>

</MenuBar>

</children>

</AnchorPane>

</children>

</StackPane>

**Database handler:**

package library.assistant.database;

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.chart.PieChart;

import javax.swing.JOptionPane;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import library.assistant.ui.listbook.BookListController.Book;

import library.assistant.ui.listmember.MemberListController;

import org.apache.logging.log4j.Level;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

public final class DatabaseHandler {

private final static Logger LOGGER = LogManager.getLogger(DatabaseHandler.class.getName());

private static DatabaseHandler handler = null;

private static final String DB\_URL = "jdbc:derby:database;create=true";

private static Connection conn = null;

private static Statement stmt = null;

static {

createConnection();

inflateDB();

}

private DatabaseHandler() {

}

public static DatabaseHandler getInstance() {

if (handler == null) {

handler = new DatabaseHandler();

}

return handler;

}

private static void inflateDB() {

List<String> tableData = new ArrayList<>();

try {

Set<String> loadedTables = getDBTables();

System.out.println("Already loaded tables " + loadedTables);

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(DatabaseHandler.class.getClass().getResourceAsStream("/resources/database/tables.xml"));

NodeList nList = doc.getElementsByTagName("table-entry");

for (int i = 0; i < nList.getLength(); i++) {

Node nNode = nList.item(i);

Element entry = (Element) nNode;

String tableName = entry.getAttribute("name");

String query = entry.getAttribute("col-data");

if (!loadedTables.contains(tableName.toLowerCase())) {

tableData.add(String.format("CREATE TABLE %s (%s)", tableName, query));

}

}

if (tableData.isEmpty()) {

System.out.println("Tables are already loaded");

}

else {

System.out.println("Inflating new tables.");

createTables(tableData);

}

}

catch (Exception ex) {

LOGGER.log(Level.ERROR, "{}", ex);

}

}

private static void createConnection() {

try {

Class.forName("org.apache.derby.jdbc.EmbeddedDriver").newInstance();

conn = DriverManager.getConnection(DB\_URL);

}

catch (Exception e) {

JOptionPane.showMessageDialog(null, "Cant load database", "Database Error", JOptionPane.ERROR\_MESSAGE);

System.exit(0);

}

}

private static Set<String> getDBTables() throws SQLException {

Set<String> set = new HashSet<>();

DatabaseMetaData dbmeta = conn.getMetaData();

readDBTable(set, dbmeta, "TABLE", null);

return set;

}

private static void readDBTable(Set<String> set, DatabaseMetaData dbmeta, String searchCriteria, String schema) throws SQLException {

ResultSet rs = dbmeta.getTables(null, schema, null, new String[]{searchCriteria});

while (rs.next()) {

set.add(rs.getString("TABLE\_NAME").toLowerCase());

}

}

public ResultSet execQuery(String query) {

ResultSet result;

try {

stmt = conn.createStatement();

result = stmt.executeQuery(query);

}

catch (SQLException ex) {

System.out.println("Exception at execQuery:dataHandler" + ex.getLocalizedMessage());

return null;

}

finally {

}

return result;

}

public boolean execAction(String qu) {

try {

stmt = conn.createStatement();

stmt.execute(qu);

return true;

}

catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Error:" + ex.getMessage(), "Error Occured", JOptionPane.ERROR\_MESSAGE);

System.out.println("Exception at execQuery:dataHandler" + ex.getLocalizedMessage());

return false;

}

finally {

}

}

public boolean deleteBook(Book book) {

try {

String deleteStatement = "DELETE FROM BOOK WHERE ID = ?";

PreparedStatement stmt = conn.prepareStatement(deleteStatement);

stmt.setString(1, book.getId());

int res = stmt.executeUpdate();

if (res == 1) {

return true;

}

}

catch (SQLException ex) {

LOGGER.log(Level.ERROR, "{}", ex);

}

return false;

}

public boolean isBookAlreadyIssued(Book book) {

try {

String checkstmt = "SELECT COUNT(\*) FROM ISSUE WHERE bookid=?";

PreparedStatement stmt = conn.prepareStatement(checkstmt);

stmt.setString(1, book.getId());

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

int count = rs.getInt(1);

System.out.println(count);

return (count > 0);

}

}

catch (SQLException ex) {

LOGGER.log(Level.ERROR, "{}", ex);

}

return false;

}

public boolean deleteMember(MemberListController.Member member) {

try {

String deleteStatement = "DELETE FROM MEMBER WHERE id = ?";

PreparedStatement stmt = conn.prepareStatement(deleteStatement);

stmt.setString(1, member.getId());

int res = stmt.executeUpdate();

if (res == 1) {

return true;

}

}

catch (SQLException ex) {

LOGGER.log(Level.ERROR, "{}", ex);

}

return false;

}

public boolean isMemberHasAnyBooks(MemberListController.Member member) {

try {

String checkstmt = "SELECT COUNT(\*) FROM ISSUE WHERE memberID=?";

PreparedStatement stmt = conn.prepareStatement(checkstmt);

stmt.setString(1, member.getId());

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

int count = rs.getInt(1);

System.out.println(count);

return (count > 0);

}

}

catch (SQLException ex) {

LOGGER.log(Level.ERROR, "{}", ex);

}

return false;

}

public boolean updateBook(Book book) {

try {

String update = "UPDATE BOOK SET TITLE=?, AUTHOR=?, PUBLISHER=? WHERE ID=?";

PreparedStatement stmt = conn.prepareStatement(update);

stmt.setString(1, book.getTitle());

stmt.setString(2, book.getAuthor());

stmt.setString(3, book.getPublisher());

stmt.setString(4, book.getId());

int res = stmt.executeUpdate();

return (res > 0);

}

catch (SQLException ex) {

LOGGER.log(Level.ERROR, "{}", ex);

}

return false;

}

public boolean updateMember(MemberListController.Member member) {

try {

String update = "UPDATE MEMBER SET NAME=?, EMAIL=?, MOBILE=? WHERE ID=?";

PreparedStatement stmt = conn.prepareStatement(update);

stmt.setString(1, member.getName());

stmt.setString(2, member.getEmail());

stmt.setString(3, member.getMobile());

stmt.setString(4, member.getId());

int res = stmt.executeUpdate();

return (res > 0);

}

catch (SQLException ex) {

LOGGER.log(Level.ERROR, "{}", ex);

}

return false;

}

public static void main(String[] args) throws Exception {

DatabaseHandler.getInstance();

}

public ObservableList<PieChart.Data> getBookGraphStatistics() {

ObservableList<PieChart.Data> data = FXCollections.observableArrayList();

try {

String qu1 = "SELECT COUNT(\*) FROM BOOK";

String qu2 = "SELECT COUNT(\*) FROM ISSUE";

ResultSet rs = execQuery(qu1);

if (rs.next()) {

int count = rs.getInt(1);

data.add(new PieChart.Data("Total Books (" + count + ")", count));

}

rs = execQuery(qu2);

if (rs.next()) {

int count = rs.getInt(1);

data.add(new PieChart.Data("Issued Books (" + count + ")", count));

}

}

catch (Exception e) {

e.printStackTrace();

}

return data;

}

public ObservableList<PieChart.Data> getMemberGraphStatistics() {

ObservableList<PieChart.Data> data = FXCollections.observableArrayList();

try {

String qu1 = "SELECT COUNT(\*) FROM MEMBER";

String qu2 = "SELECT COUNT(DISTINCT memberID) FROM ISSUE";

ResultSet rs = execQuery(qu1);

if (rs.next()) {

int count = rs.getInt(1);

data.add(new PieChart.Data("Total Members (" + count + ")", count));

}

rs = execQuery(qu2);

if (rs.next()) {

int count = rs.getInt(1);

data.add(new PieChart.Data("Active (" + count + ")", count));

}

}

catch (Exception e) {

e.printStackTrace();

}

return data;

}

private static void createTables(List<String> tableData) throws SQLException {

Statement statement = conn.createStatement();

statement.closeOnCompletion();

for (String command : tableData) {

System.out.println(command);

statement.addBatch(command);

}

statement.executeBatch();

}

public Connection getConnection() {

return conn;

}

}

**SYNOPSIS**

The main objective of this “LIBRARY MANAGEMENT SYSTEM” project is to develop the Software for the library which requires the information management of the books and members in the library. This project work aim to design and implement a computerized library management system.

It has a facility of admin login through which the admin can monitor the whole system. In this computerized system where we can record various transactions like issue of books, return books, addition of members. It also has facility of an online notice board where members are not renew or submission of book. 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.

Overall this project of ours is being developed to help members and librarians to maintain the library in the best way possible and also reduce the human works.

****

I am **SHYAM SUNDAR S** (17BCT046) doing final year Computer Technology in PSG College of Arts and Science, Batch 2017 – 2020.