National College of Computer Studies

Paknajol,Kathmandu



Tribhuvan University

Submitted By:                                                  Submitted to:

Bishal Phuyal                              Mr.Bimal Lamichhane

Prabin Buddhacharya                 Mr.Ujjwol Ghimre

Ravi Singh

Sanish Maharjan

Sijan Maharjan

**Abstract**

This system is entitled "**Library Management System**". The main objective of this system is to manage the information in the library regarding the borrow, return, reservation, fine payment, etc.

In this system, books are recorded based on the title and authors and category. The copies of the same books that the library has are also recorded in the system. Borrowers need to be registered in the system for them to borrow any books. The member id is to be recorded in the system when members borrow any book with the borrow date with submission period included. In case the books the borrower is requesting are out of stock, the reservation for that particular book is done to provide when the books are in stock or when the required book is returned by another fellow borrower. When a borrower crosses the limitation time to return the book, the fine payment is also included in the system. In case the book is returned, the information is recorded in the system.

**Group Members:**

Sijan Maharjan

Sanish Maharjan

Prabin Buddhacharya

Ravi Singh

Bishal Phuyal

**Acknowledgement**

We would like to express our special thanks of gratitude to our teacher Mr. Bimal Lamichhane and Mr. Ujjwol Ghimire  as well as our principal Ganesh Basnet who gave us the opportunity to do this wonderful project on the topic “**Library Management System**”, which also helped us in doing a lot of research and coding and we came to know about so many new things we are really thankful to them. We would like to thank our Java teacher Mr.Sumit Ghising for teaching Java in such an interactive and understandable way which helped a lot in building this project.Other Personnel who we should not forget all the honourable Sirs and Madams from the administrations managing extra classes on mysql online regarding the current situations, we find ourself grateful for them.

Secondly We would also like to thank our other classmates who came up with their own ideas and helped us a lot in finalizing this project within the limited time frame.

Lastly, thanks to all those websites and platforms from where we could solve our queries easily. Platforms like Github, Zoom Video Conferencing helped the members to be collaborative and work as a team. And services like google docs and slides helped to document the project. So we express our gratitude for having  all those platforms.

Table of Content

[Chapter:1 Introduction](#_heading=h.3znysh7) **1**

[Background](#_heading=h.x49ey5x2r3bg) 1

[Purpose](#_heading=h.pkxubt33goou) 2

[Function](#_heading=h.tyjcwt) 2

[**Chapter 2: Diagrams**](#_heading=h.71eex3pq1t3u) **3**

[ER Diagram:](#_heading=h.3dy6vkm) 3

[Data Flow Diagram (DFD)](#_heading=h.gpv36bgnhr8y) 4

[DFD Level 0:](#_heading=h.1t3h5sf) 4

[DFD level 1:](#_heading=h.sk5175lc32iu) 4

[DFD level: 2](#_heading=h.fiypd1jtqmlc) 6

[**fig:1.3 (Data Flow Diagram Level 0)**](#_heading=h.gg26tkeu4xby) **6**

[**Chapter:3 Tables of database:**](#_heading=h.s8odw7x51vhf) **7**

[**Table:1.1 (Table for Book\_Table in database and its relations)**](#_heading=h.e9r7chjvj21d) **7**

[**Chapter:4 JAVA**](#_heading=h.swhxdqw1jpz6) **10**

[Main Application](#_heading=h.sdcnyf3c14oa) 10

[**Database Connection**](#_heading=h.g19mkbyweret) **10**

[Packages](#_heading=h.7h5hb6ti9akf) 11

[Controller](#_heading=h.gui6qcof422) 11

[Model](#_heading=h.xsd5eet8qinq) 12

[repository](#_heading=h.rkualnpjr23f) 14

[view](#_heading=h.jlb89kzhvm2m) 15

[This is used to view the datas in the console of the user.](#_heading=h.c8a68c6ijo7p) 15

[**Conclusion**](#_heading=h.xml1mig4nofo) **16**

[**Reference and Tools**](#_heading=h.y4hsl1gzds07) **16**

# Chapter:1 Introduction

Library Management System is the java based database management system that helps Library to manage the Readers, Books, Writer in their database. This system is built in such a way that it helps the librarian to clear up their head by organizing all the datas related to the library. This system is especially made to manage the datas of the community  library. This is the era where all the registered data is slowly getting  transferred to the computer and internet. Transferring the humongous datas of the library is a hectic job but to fulfill this gap this system comes handy. This system is backed by modern technologies and methods  which stores and helps access the data in a more convenient way.This database program is made focusing the librarian to manage the book of the library that is borrowed and reserved and available in the library.

The System has implemented the MVC design pattern which makes the program more efficient for both users(librarian and borrower) and the developer. The code is written with the universal coding styles and code is so manageable that it is easily trackable to the maintenance team using technical documents. Though we wrote about the following terms in the following document these are the technical and JIC cases but the main advantage and the backbone of the system that makes our product so demanding to be acquired is the database part. We have done no compromise on the technology used in the backend part. Our Mysql team has made successful work on bringing the most logical, space saving and fast queries on the table.

## Background

Managing the datas of the library is a hard call before the Library management system.Different problems occur while managing datas of the library before. Such as errors in data collection,loss of books,loss of data collection etc. So we made a “Library Management System” to tackle all these problems which helps librarians to manage the library using a computerized system where he/she can add new books and control and manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

Acquiring this system, the library has gained more security than ever before. With this system there will be no loss of book record or member record which generally happens when a non computerized system is used. Books and members modules are also included which helps in keeping track of members using books of the library and also contains detailed information of books that are in the library.This system is also user friendly which helps users to get detailed information about the books and also to borrow these books without their physical appearance .

## Purpose

Our main goal to make this project is remove all the unwanted mountains of registers in any offices. So we start with the library. And the bigger picture is to cover every area of life to connect with the beautiful world of computers and internet. The first step was to build the basic library management system.

This system is made for the Librarian to keep record of books in a library. This system is made for community organizations or schools to keep track of books borrowed.People can get membership and get access to the book.Basically this System is build to fulfill the requirements of the Non Profit Libraries so the transaction between money through the system is not yet build but if needed the code is build in such a manageable way that it can be modified for any kind of services easily and less costly.

## Function

The system is built in such a secure way that only authorized personnel can only get into the system and make changes into it. Data is shared in such a way that both librarian librarians have the full authority of the system. Librarians have to login with their ID or gmail to get access. In this system, Librarian can add and delete the books of the system.Members can check the books, search the available books, borror and reserve the book.Member can set the date of submission.There is a fine facility too, if the date of submission is due.

## 

# Chapter 2: Diagrams

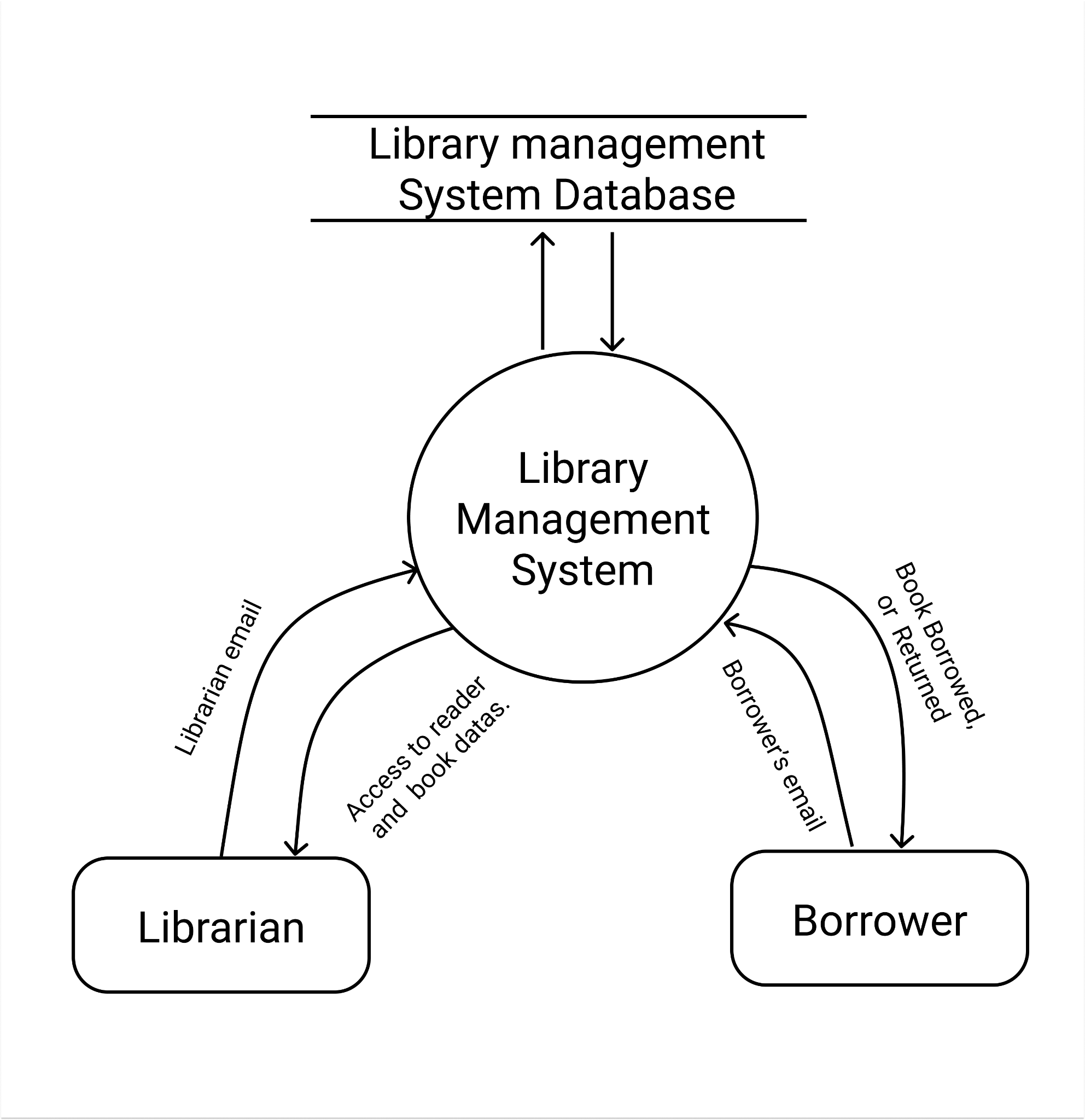
## ER Diagram:

## 

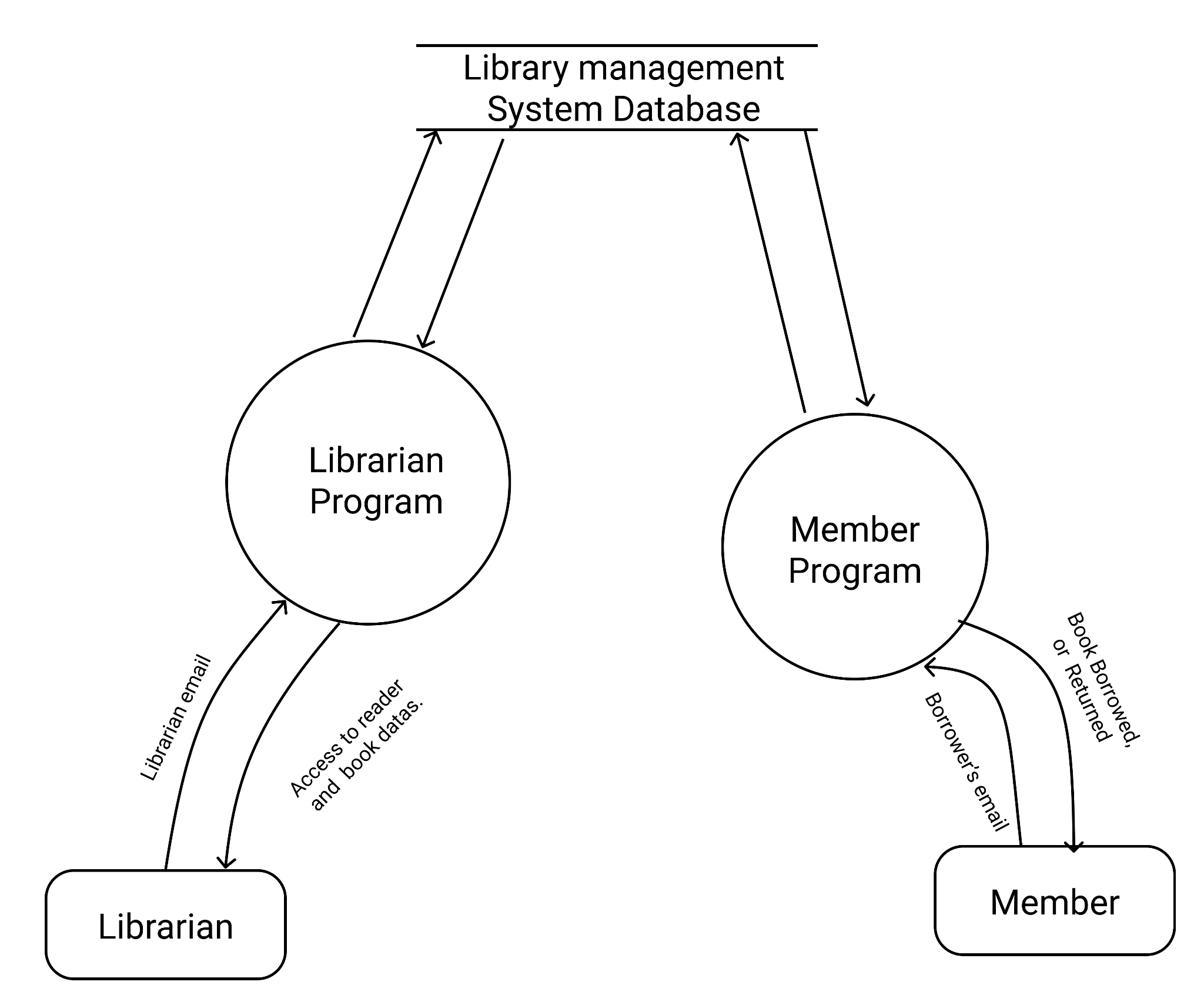
## 

## Data Flow Diagram (DFD)

#### DFD Level 0:



#### DFD level 1:



*fig:1.1(Data Flow Diagram Level 0)*

#### DFD level: 2

# 

# *fig:1.3 (Data Flow Diagram Level 0)*

# Chapter:3 Tables of database:

**BOOK:**

| ID  **<PK>**  VARCHAR(10) | Title VARCHAR(20) | ISBN  INT | CategoryId **<FK>** VARCHAR(10) | PublicationDate  DATE | Copies  INT |
| --- | --- | --- | --- | --- | --- |
| 1 | Summer Love |  | 1 | 2067 | 12 |
| 2 | Saya |  | 2 | 2070 | 14 |
|  |  |  |  |  |  |

# *Table:1.1 (Table for Book\_Table in database and its relations)*

**BOOK\_AUTHOR:**

| Book\_ID  **<FK>**  VARCHAR(10) | Author\_ID  **<FK>**  VARCHAR(10) |
| --- | --- |
| 1 | 1 |
| 1 | 2 |
| 2 | 3 |

Table:1.2 (Table for Book\_Author in database and its relation )

**AUTHOR:**

| ID  **<PK>**  VARCHAR(10) | FirstName  VARCHAR(20) | LastName  VARCHAR(20) |
| --- | --- | --- |
| 1 | Bishal | Phuyal |
| 2 | Sijan | Maharjan |
| 3 | Subin | Bhattarai |

Table:1.3 (Table for author in database and its relation )

**CATEGORY:**

| ID  **<PK>**  VARCHAR(10) | Name  VARCHAR(10) |
| --- | --- |
| 1 | Philosophy |
| 2 | Biography |
| 3 | Fiction and Fantasy |
| 4 | Science and Tech |
| 5 | Business |
| 6 | Psychology |
| 7 | Horror and Thriller |
| 8 | Adventure |
| 9 | Health |
| 10 | History |

Table:1.4 (Table for category in database and its relation )

**BORROWER:**

| Id  **<PK>**  VARCHAR(10) | Name  VARCHAR(20) | Age  INT | Address  VARCHAR(20) | Contact  INT | Email  VARCHAR(20) | MembershipDate  DATE |
| --- | --- | --- | --- | --- | --- | --- |
| BOR | Sanish Maharzan | 17 | Dallu | 9869198877 | sanz@gmail.com | 2076/12/13 |
| BOR | Prabin Buddha | 20 | Swoyambhu | 9848043780 | prbn4@gmail.com | 2077/12/14 |

Table:1.5 (Table for Borrower in database and its relation )

**BORROWED\_BOOK:**

| ID  **<PK>**  VARCHAR(10) | BOOK\_ID  **<FK>**  VARCHAR(10) | BORROWER\_ID  **<FK>**  VARCHAR(10) | BORROWED\_DATE  DATE | SUBMISSION\_DATE  DATE |
| --- | --- | --- | --- | --- |
| BORB |  |  |  |  |
| BORB |  |  |  |  |

Table:1.6 (Table for Borrowed\_book in database and its relation )

**RETURNED\_BOOK:**

| BORROWED\_BOOK\_ID  **<FK>**  VARCHAR(10) | SUBMITTED\_DATE  DATE |
| --- | --- |
|  |  |
|  |  |

Table:1.7 (Table for Returned\_Book in database and its relation )

**RESERVATION:**

| ID  **<PK>**  VARCHAR(10) | BOOK\_ID  **<FK>**  VARCHAR(10) | BORROWER\_ID  **<FK>**  VARCHAR(10) | RESERVED\_DATE  DATE |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Table:1.8 (Table for Reservation in database and its relation )

# **Chapter:4 JAVA**

## Main Application

We have the main application for the system that enters the program.

public static void main( String[] args ) throws SQLException

{

LogInView login = new LogInView();

login.displayAccounts();

}

# 

# Database Connection

It is used to connect the program with the remote server we have shared common in the cloud. It uses the methods from the packages of java.Sql which comes with the maven while starting the project.

static Connection connection;

public static Connection getConn() throws SQLException

{

if(connection == null)

connection = DriverManager.getConnection("jdbc:mysql://sql6.freesqldatabase.com:3306/sql6436752","sql6436752","a7bfimb17U");

return connection;

}

}

## Packages

These are the main packages we have as the program requires. We have basically four packages to fulfill their own needs.

### **Controller**

It controls the program that can be accessed by both librarian and browerrer. Here is the example of one of the programs main application under controller package.

BorrowedBookListRepository borrowedbooklistrepository = new BorrowedBookListRepository();

public void displayLibrarianBorrowedBookList() throws SQLException

{

List<BorrowedBookList> borrowedbooklists = borrowedbooklistrepository.getBorrowedBookListLibrarian();

for(BorrowedBookList borrowedbooklist:borrowedbooklists)

{

System.out.println("");

System.out.println(borrowedbooklist.toStringLibrarian());

}

}

### 

### 

### 

### 

### 

### 

### 

### 

### M**odel**

It stores the basic data format that may or may not be accessed by two users of the program; they are borrowers and librarians.

nt id;

String firstname;

String lastname;

public Author(int id, String firstname, String lastname)

{

this.id = id;

this.firstname = firstname;

this.lastname = lastname;

}

public int getId()

{

return id;

}

public String getFirstName()

{

return firstname;

}

public String getLastName()

{

return lastname;

}

### 

### 

### 

### 

### **repository**

BookListRepository booklistrepository = new BookListRepository();

public List<Author> getAllAuthor() throws SQLException

{

Connection connection = DBConnection.getConn();

Statement statement = connection.createStatement();

ResultSet rs = statement.executeQuery("SELECT \* FROM author");

List<Author> authors = new ArrayList<>();

while(rs.next())

{

Author author = new Author(rs.getInt(1), rs.getString(2), rs.getString(3));

authors.add(author);

}

return authors;

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **view**

### This is used to view the datas in the console of the user.

### 

BookListRepository booklistrepository = new BookListRepository();

public void displayAllBookList() throws SQLException

{

List<BookList> booklists = booklistrepository.getAllBookList();

System.out.println("\nDisplaying Book Lists...");

for(BookList booklist: booklists)

{

System.out.println();

System.out.println(booklist.toStringBorrower());

}

}

public void displayBookListByBookName(String booktitle) throws SQLException

{

List<BookList> booklists = booklistrepository.getBookListByBookName(booktitle);

System.out.println("\nDisplaying Book Lists Titled '" + booktitle + "'");

for(BookList booklist: booklists)

{

System.out.println();

System.out.println(booklist.toStringBorrower());

}

}

For the further detail of the system you can see the Technical document. We have not included much to maintain the simplicity of the report.

# Conclusion

Traditional ways and its flawsData Loss and time consuming.Used the best logic in the database and the program.Used the latest technologies and best practises to make users friendly.Business Logic Part. Mostly for Non Profit. Code for Money transaction is not included.But can be expanded to commercial if needed..

# Reference and Tools

Tools used for this project are:

1. Google docs and slides for documentation
2. Figma for designing DFD and ERD
3. Github/Gitlab for java codes Version control
4. Eclipse, Netbeans and VSCode as main IDE.

Referencing:

1. Tutorialspoint: <https://www.tutorialspoint.com/java/index.htm/>
2. Tutorialspoint: <https://www.tutorialspoint.com/jdbc/jdbc-insert-records.htm/>
3. MVC:<https://www.tutorialspoint.com/mvc_framework/mvc_framework_first_application.htm/>
4. GeeksforGeeks: <https://www.geeksforgeeks.org/introduction-to-jdbc/>
5. GeeksforGeeks: <https://www.geeksforgeeks.org/interfaces-in-java/>
6. StackOverFlow:<https://stackoverflow.com/questions/52731368/cannot-connect-to-mysql-on-localhost-using-jdbc-driver/52731930/>
7. Stackoverflow:<https://stackoverflow.com/questions/23272500/java-cant-connect-to-jdbc/>
8. Youtube Videos for understanding MVC, Java and SQL and other websites.