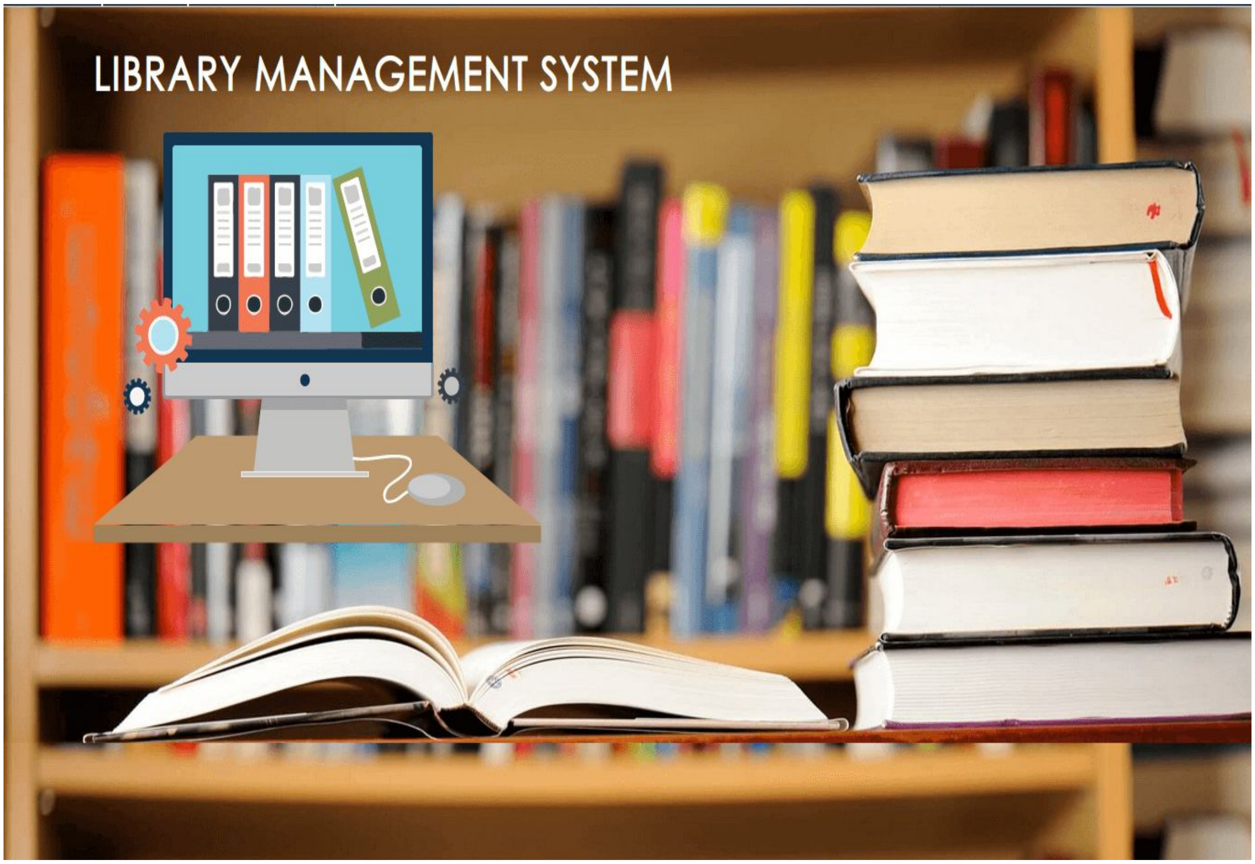
DBMS PROJECT REPORT

on

LIBRARY MANAGEMENT SYSTEM



SUBMITTED BY :-

K. Durga Prasad Reddy - 111501012
M. Aditya - 111501016

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 in normal library management systems like facility of user login and a facility of teachers login . 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 , The student can also check for the fine obtained after returning the book in the library and the students can request the librarian to add new books by filling the book request form. The student has the privilege to check his details and change them if required. This system will also help the faculty to keep track of all the books or records issued by the users and corresponding fine .

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.

INTRODUCTION :

Database Management System has been one of the most used CSE terms these days. The answer to what exactly it is and how it makes maintaining records easy was first given by Edgar Codd. DBMS is a computer software application that interacts with end-users, other applications, and the database itself to capture and analyze data. Before DBMS, normal File System was used to store data and access it. The advantages of DBMS over File System like **Data Independence, Efficient Data access, Data Integrity, Data Administration, Concurrent Access, Crash recovery, Reduced Application development time** makes this most used software.**DBMS** is used in diversified fields like in banking, Air-lines, Universities, Finance etc and can be integrated with Machine Learning like in OtterTune for heavy data handling.

As per our plan, we designed a Database Management System for Library Management purposes which satisfies almost all the requirements that are specified in the Required Management section. The Design of the Project is Shown as ER diagram in the DESIGN section and local schema for the Design was Written using user-friendly software **mariaDB** as back-end RDBMS(The details of the tables used will be in the design section), **MySQL** as Query language and **HTML, CSS, PHP** as front-end to design the software.

REQUIREMENT ANALYSIS :

The database that we have designed is meant to satisfy the requirements that we have specified below :

- There are two types of members first users and second the employees
- The employees are further divided in four categories
  - Librarian
  - Assistant Librarian
  - Manager
  - Janitor
- Library consist of four departments:
  - Acquisition department.
  - Circulation Department.
  - Reference Department.
  - Technical Department.
- Each user has the privilege to access to his/her information and the user can even change the information if he/she wants.
- Each user can view the books issued by him/her.
- The user can issue the books only in the presence of assistant librarian or librarian.
- One or more managers are assigned to each departments.
- The user an issue books until he reaches the maximum issued books limit and after that he/she has to return the book back to library.
- List of all books issued by a member has to be recorded in the database with Date of issue and has to return it within four weeks, otherwise fine will be

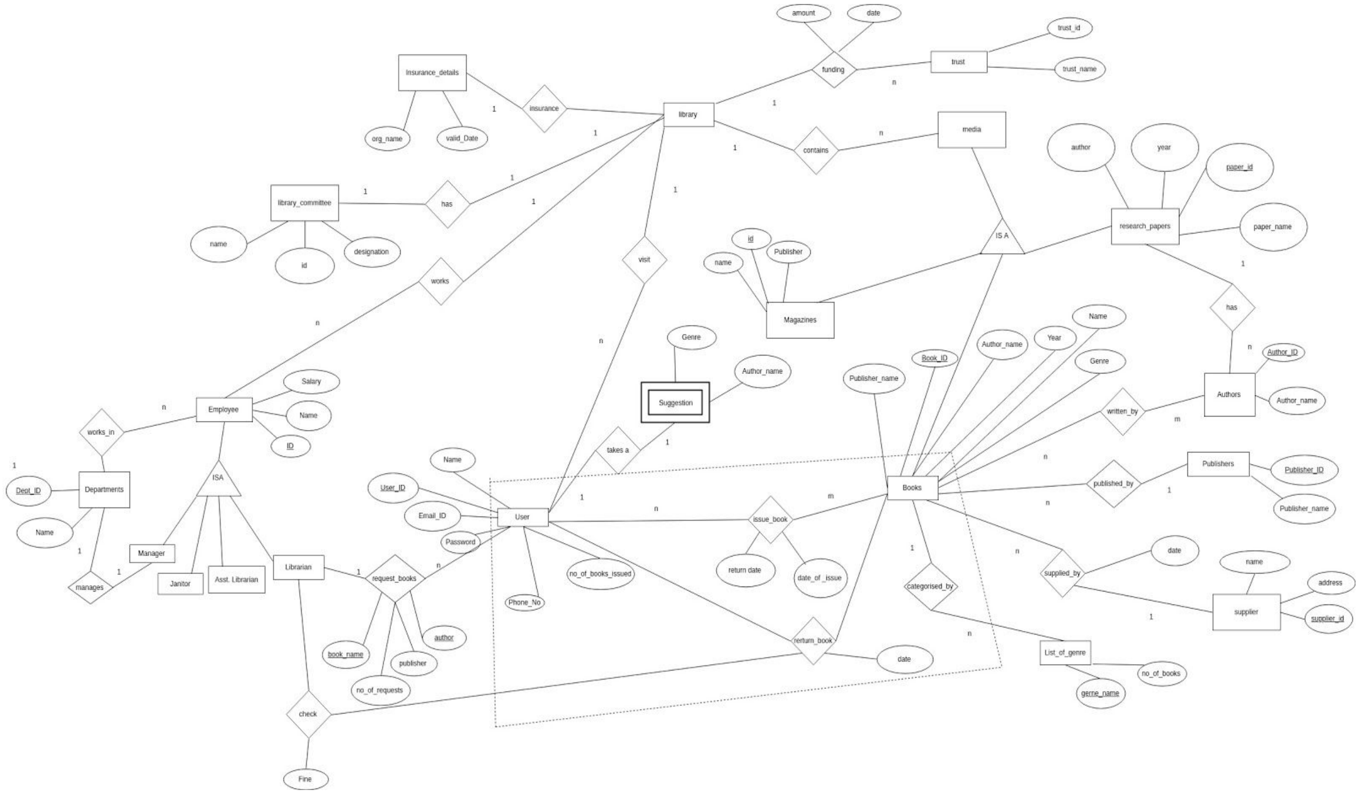
imposed and until all the fine are due, a member is not allowed to issue any book.

- ➔ The users can return the book only in presence of assistant librarian or librarian.
- ➔ Fine will be imposed if the user return the book after the due date (We are thinking the fine will be Rs. 10 for each day after the Due date).
- ➔ The Members can only have the read access for the books or magazines in the database.
- ➔ In case of lost book or damaged book, the user has to pay a fine equal to the original cost of the book.
- ➔ Information about the Publishers and Suppliers are stored .
- ➔ A member can request the librarian to buy a new book, if there are more than a certain number of requests on that then librarian asks owner to buy the book.
- ➔ Librarian has to classify books based on genre, author and publication.\
- ➔ Top magazines which are classified based on genre, newspapers of all language should also be recorded in the database and members can have access to them.
- ➔ Librarian and heads of each department are members of the Library committee which takes important decisions, all this information is to be recorded in the database.
- ➔ Trust funds from Private and Government Organizations are also recorded in the database.
- ➔ Insurance information is also recorded.
- ➔ The User should be granted by the librarian for an account in that library.
- ➔ When the member is logged into his account, he is asked his interests, favorite author and publications and based on that different books are suggested.
- ➔ Member can upgrade his membership from free to premium by paying some money, premium user can access any book while free user are restricted to view only few.

DATABASE DESIGN :

ER - DIAGRAM :

The ER-Diagram provides below tells the Design of Library Management system Database design that we give a rough idea about what our Database system will be like.



LOGICAL SCHEMA :

The logical schema for The tables used in the database are provided below. Here i have given only the basic glimpse of what our database consists of for further details like primary keys, foreign keys, constraints, datatypes please refer to the following link directed to the SQL file(tables) :

1. research_papers (paper_id,paper_name,corresponding_author,year)
2. Authors (author_id,author_name)
3. Genres (genre_name)
4. Paper_author (author_id,paper_id)
5. trusts(organosation_id,organization_name)
6. Publishers(publisher_name,publisher_name)
7. books(ISBN,book_name,year,corresponding_author,publisher_id,genre_name,price,_count)
8. book_author(ISBN,author_id)
9. suppliers(supplier_id,supplier_name)
10. supplier_book(ISBN,supplier_id,supplied_date)
11. magazines(magazine_id,magazine_name,publisher_id)
12. funding(organization_id,amount,transfer_date)
13. employees(employee_id,employee_name,username,_password,designation,salary)
14. users(user_id,_password,name,phone_no,email_address,issued_books)
15. issue_book(id,ISBN,user_id,issue_date,return_date,status_return)
16. departments(department_id,department_name)
17. manages(employ_id,department_id)
18. return_book(id,return_book)
19. check_fine(id,fine,paid)
20. request_book(user_id,book_name,corresponding_author)
21. request_book_2(book_name,corresponding author,publisher_id,_count)
22. insurance_details(id,org_name,valid_till)
23. library_committee(id,name,designation)
24. lost_book(user_id,ISBN)

User Roles and Privileges :

The privileges of various type of members are roughly specified below :

1. **Users** : The users have the privilege to see all the research papers, magazines, books, publisher details, authors details available in the library. They even have the privilege to filter books based on their Genres, book_name, author name, publisher name etc. the user has the privilege to access the details corresponding to that user like profile details, books issued, book returned, fine details. The user also has the privilege to request a library for the book that is not available in the library.
2. **Manager** : The managers have the privilege to insert, update, delete books, authors , publisher, suppliers. The manager has the privilege to see his profile information and change it if required. To give a simple understanding Manages only manage the data related to books or research papers, magazine.
3. **Assistant librarian** : The assistant librarians have the privilege to see all the books, publishers, authors etc. They also have the privilege to access user information. They have the privilege to issue the book to the user, accept return book from user, and take new book requests from users. They even have the privilege to check the fine information about the users. They can even have the access to their own information.
4. **Librarian** : The librarians are the head of the library and have the access to all the information in the library. Only librarian has the access to informations about funds, insurance, trusts etc. which will main the library with the help of external organizations.

PROCEDURES AND TRIGGERS :

This sections contains the triggers and procedures that we have implemented in our database. This define s the core of our database system. The Procedures a are a set of statements in sql which provides all the statement that have to be executed for a required functionality .

A trigger is a special type of stored procedure. It is special because it is not called directly like a stored procedure. The main difference between a trigger and a stored procedure is that a trigger is called automatically when a data modification event is made against a table whereas a stored procedure must be called explicitly.

Normalization :

Normalization Is a systematic way of ensuring that a database structure is suitable for general-purpose querying and free of insertion, updation and deletion anomalies- that could lead to loss in data integrity. It is important to ensure that there is a certain level of normalization in th database.

Here in this database we have decomposed the database such that it is in 3NF. This was the first step because the procedures and triggers depends upon the table definition. (i.e as the table definitions changes triggers and procedures may also change)

Procedures :

The various procedures that we have used in our database are listed below along with their functionality and their input and output :

Procedure name	**Functionality**
Get_book_history	Takes an book ISBN number as the input and return the names of all the users issued that book
total_fine	Takes an user_id as an input and returns the total fine for that particular users.
create_user	Takes username and password of a user and create user for that username and password and also assigns the roles and privileges to that users using Dynamic SQL
create_user_employ	Takes the username , password and designation of the employee and creates a user for that according giving all the roles and privileges to that employees as per their designation
user_fine_details	No input is taken this procedures tells the details of all the books issued by the user the output table contains(id,ISBN,user_id,issue_date,fine,paid_status)
req_book	This procedures takes (user_id,bk_name,corr_author as input and if the book_name corresponding to the corresponding author is already present in the reuest_book table and updates the count if t exists else inserts it.
publisher_books	Takes the publisher name as input and returns all the books published by that corresponding publisher
author_book	Takes the author name as input and returns all the books written by that particular author.
book_genre	Takes the genre as an input and returns all the books of that particular genre.
user_details	This takes no input an can only be accessed by the user to get all the details of that user
user_book_details	This takes no input and can only be accessed by the user to

	get the list of all the books issued by that user.

Triggers :

Trigger name	**Functionality**
Valid_issue	This is executed before there is an insert in the issue_book table (which stores the details of the books issued). This trigger checks all the constraints and return an error if the user is not eligible to issue a book. The constraints are like 1) Check if user has issue maximum books(i.e 5) 2) No. of books lost or damaged > 3 3) Whether books are available in library or not
book_lost	This trigger is executed before insertion on lost_book table. This trigger first checks whether the user has issued the book or not if the user has issued it then it will insert the fine in the check_fine table and decrement the issued_books count in users table
fine_cal	The trigger is executed after there is an insert in the return_book tabel. To be precise it actually calculate the fine using the difference between the deadline if the book and the return date of the book
check_issued_book	Before an update is made on the user table this trigger scheck where ther the issued_book field count is correct or not just to be on the safe side
ret_book	Before insert in the return_book table the trigger checks whether the book is actually issued by the user or not if the book is not issued by the user it prints an error else it decrement the issed_books count on users table by one and increment the books_count in books table by one
Valid iss_2	Before insertion on the issue_book table it actually checks whether the user has already issued that book or not and also checks whether the total fine has exceed 500 or not.
iss_book	After inserting in the issue_book table it incements the issued_books count in users table by 1 and decrements the _count in book table by 1.

check_username	Before insertion on the users table this trigger checks whether that user_id is already taken in the employ table or not. Since user-id must be unique for all people(users and employees)
check_user_id	Before insertion on the employee table it actually checks whether the username already exists in the user table or not.
check_isbn	Checks whether the ISBN number is valid or not i.e greater than 0 or not

RESULTS AND CONCLUSION :

FAULTS COMPARED TO REQUIREMENT ANALYSIS :

The database we created couldn’t meet some of the features listed in requirement analysis. They are :

1. **When the member is logged into his account, he is asked his interests, favorite author and publications and based on that different books are suggested.**

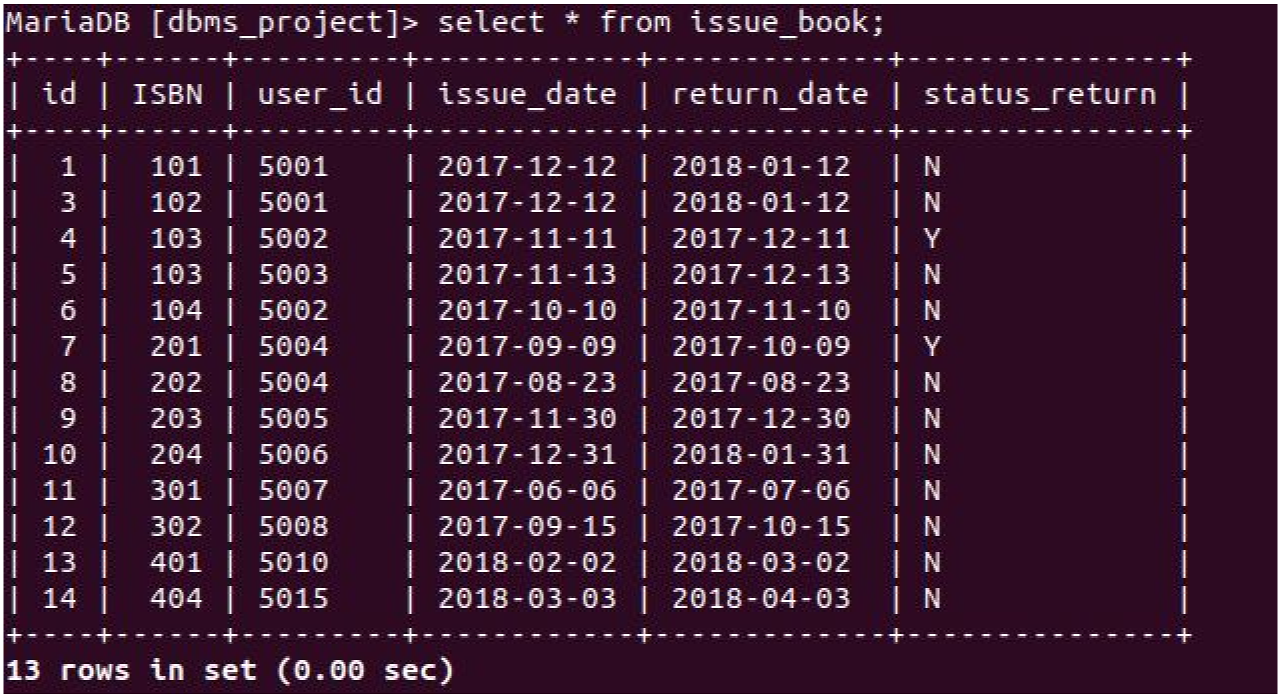
➔ We were not able to do this because in order to do this it would require an object oriented approach which is not supported in mysql.

2. **Member can upgrade his membership from free to premium by paying some money, premium user can access any book while free user are restricted to view only few.**

➔ We tried to do this for our database but we were not able to achieve it as it requires a deeper understanding of mysql and mariadb database. So we thought of omitting this requirement from our database.

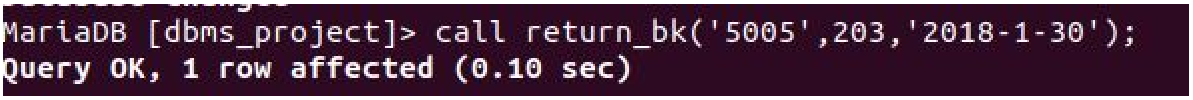
RESULTS :

1. If a book is returned in the library the fine is upgraded in the check_fine table and status_returnis changed in the issue_table:
   - ➔ Initial issue_book table :



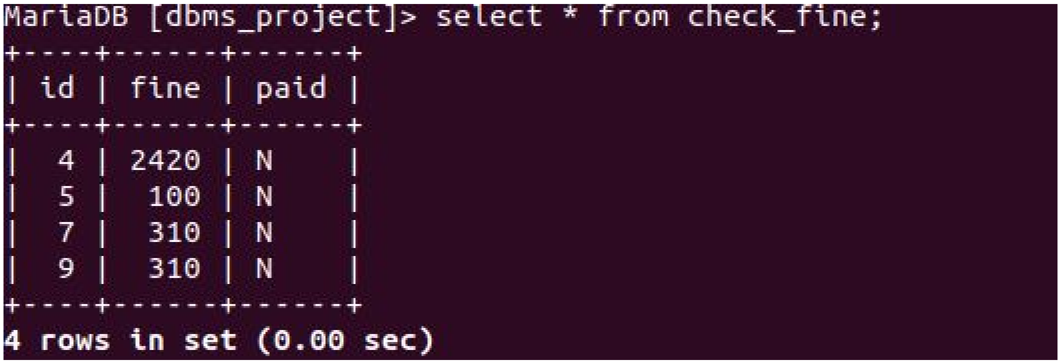
```
MariaDB [dbms_project]> select * from issue_book;
+----+------+---------+------------+-------------+---------------+
| id | ISBN | user_id | issue_date | return_date | status_return |
+----+------+---------+------------+-------------+---------------+
|  1 |  101 | 5001    | 2017-12-12 | 2018-01-12  | N             |
|  3 |  102 | 5001    | 2017-12-12 | 2018-01-12  | N             |
|  4 |  103 | 5002    | 2017-11-11 | 2017-12-11  | Y             |
|  5 |  103 | 5003    | 2017-11-13 | 2017-12-13  | N             |
|  6 |  104 | 5002    | 2017-10-10 | 2017-11-10  | N             |
|  7 |  201 | 5004    | 2017-09-09 | 2017-10-09  | Y             |
|  8 |  202 | 5004    | 2017-08-23 | 2017-08-23  | N             |
|  9 |  203 | 5005    | 2017-11-30 | 2017-12-30  | N             |
| 10 |  204 | 5006    | 2017-12-31 | 2018-01-31  | N             |
| 11 |  301 | 5007    | 2017-06-06 | 2017-07-06  | N             |
| 12 |  302 | 5008    | 2017-09-15 | 2017-10-15  | N             |
| 13 |  401 | 5010    | 2018-02-02 | 2018-03-02  | N             |
| 14 |  404 | 5015    | 2018-03-03 | 2018-04-03  | N             |
+----+------+---------+------------+-------------+---------------+
13 rows in set (0.00 sec)
```

   - ➔ After returning book:



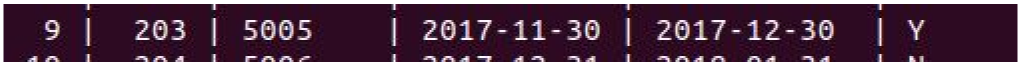
```
MariaDB [dbms_project]> call return_bk('5005',203,'2018-1-30');
Query OK, 1 row affected (0.10 sec)
```

   - ➔ Fine is automatically inserted in the table for that issue_id (ie. id)



```
MariaDB [dbms_project]> select * from check_fine;
+----+------+------+
| id | fine | paid |
+----+------+------+
|  4 | 2420 | N    |
|  5 |  100 | N    |
|  7 |  310 | N    |
|  9 |  310 | N    |
+----+------+------+
4 rows in set (0.00 sec)
```

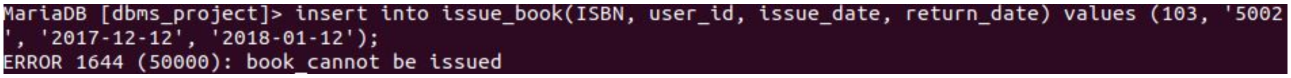
➔ Return_status in issue book for that corresponding id is updated to ‘Y’;



```
  9 |   203 | 5005     | 2017-11-30 | 2017-12-30  | Y
```

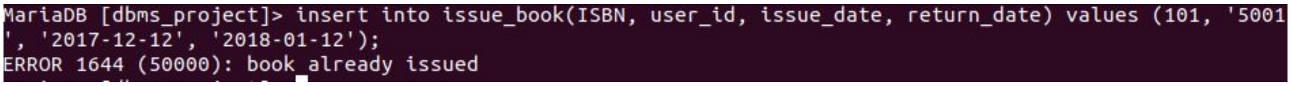
2. When book is issued by the user with fine greater than 500 the :

- Insert value into issue book



```
MariaDB [dbms_project]> insert into issue_book(ISBN, user_id, issue_date, return_date) values (103, '5002
', '2017-12-12', '2018-01-12');
ERROR 1644 (50000): book cannot be issued
```

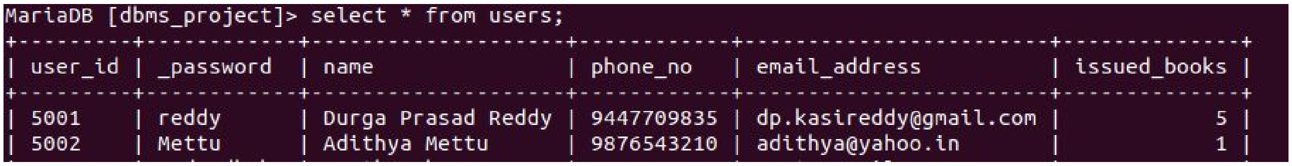
3. When a user try to issue a book that is already issued by that user:



```
MariaDB [dbms_project]> insert into issue_book(ISBN, user_id, issue_date, return_date) values (101, '5001
', '2017-12-12', '2018-01-12');
ERROR 1644 (50000): book already issued
```

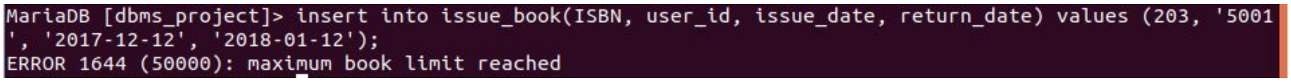
4. When a user try to issue a book when he has already issued books = 5:

➔ Issued_books :



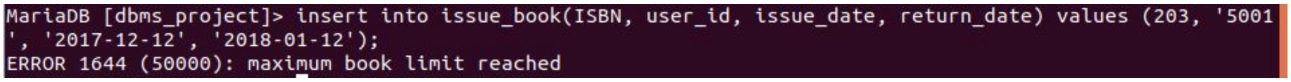
```
MariaDB [dbms_project]> select * from users;
+---------+-----------+--------------------+------------+------------------------+--------------+
| user_id | _password | name               | phone_no   | email_address          | issued_books |
+---------+-----------+--------------------+------------+------------------------+--------------+
| 5001    | reddy     | Durga Prasad Reddy | 9447709835 | dp.kasireddy@gmail.com |            5 |
| 5002    | Mettu     | Adithya Mettu      | 9876543210 | adithya@yahoo.in       |            1 |
```

➔ Issuing book



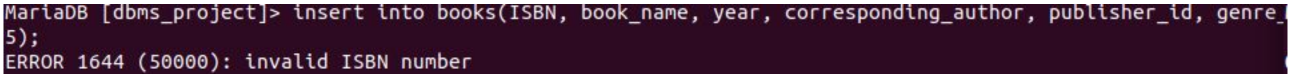
```
MariaDB [dbms_project]> insert into issue_book(ISBN, user_id, issue_date, return_date) values (203, '5001
', '2017-12-12', '2018-01-12');
ERROR 1644 (50000): maximum book limit reached
```

5. When the user is trying to issue the book which is not available :



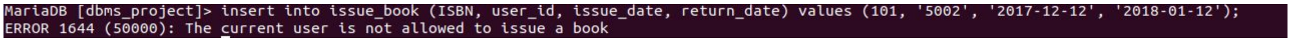
```
MariaDB [dbms_project]> insert into issue_book(ISBN, user_id, issue_date, return_date) values (203, '5001
', '2017-12-12', '2018-01-12');
ERROR 1644 (50000): maximum book limit reached
```

6. When we try to insert a new book with negative ISBN number :



```
MariaDB [dbms_project]> insert into books(ISBN, book_name, year, corresponding_author, publisher_id, genre_
5);
ERROR 1644 (50000): invalid ISBN number
```

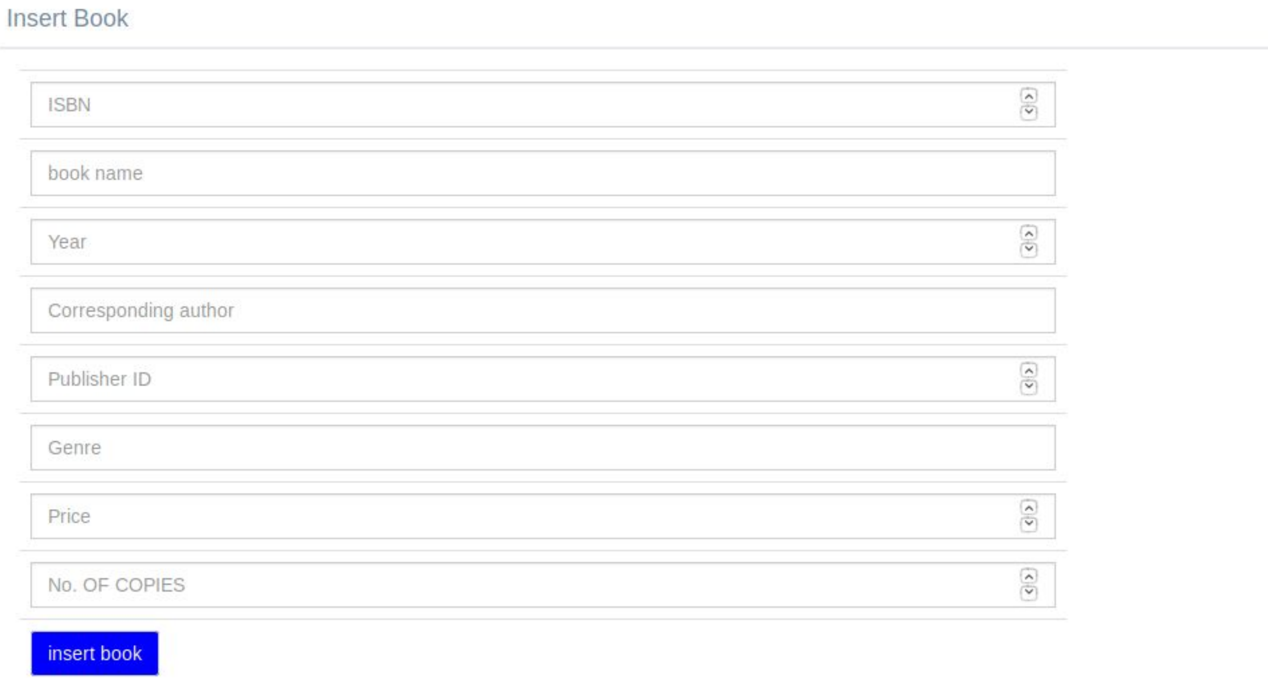
7. When a user who don't have privilege to issue book try to issue the book :



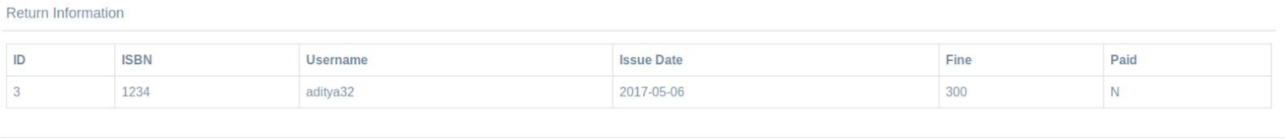
```
MariaDB [dbms_project]> insert into issue_book (ISBN, user_id, issue_date, return_date) values (101, '5002', '2017-12-12', '2018-01-12');
ERROR 1644 (50000): The current user is not allowed to issue a book
```

USER INTERFACES :

1. User Interface for Inserting books:



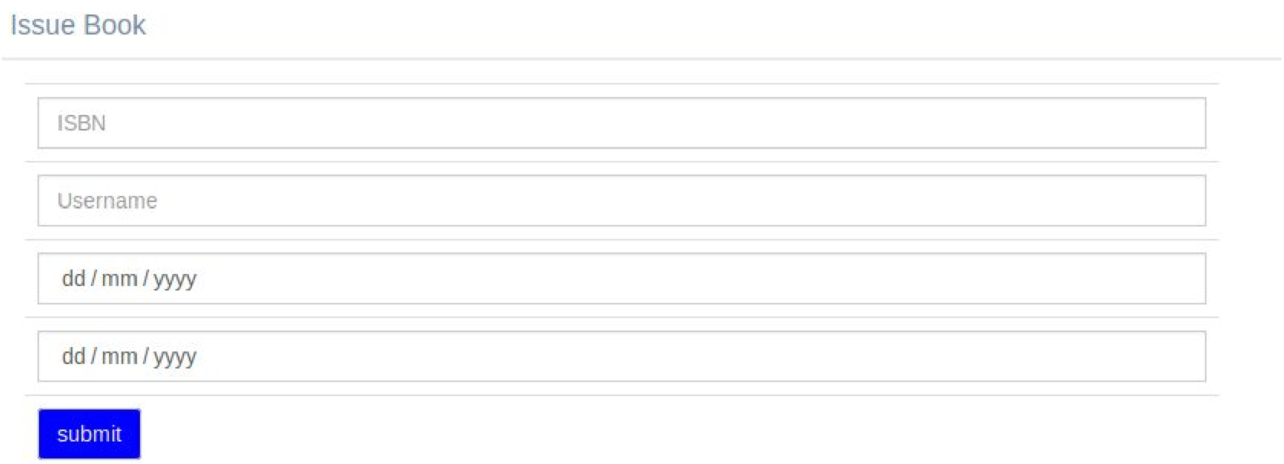
2. Information about books that have been returned :



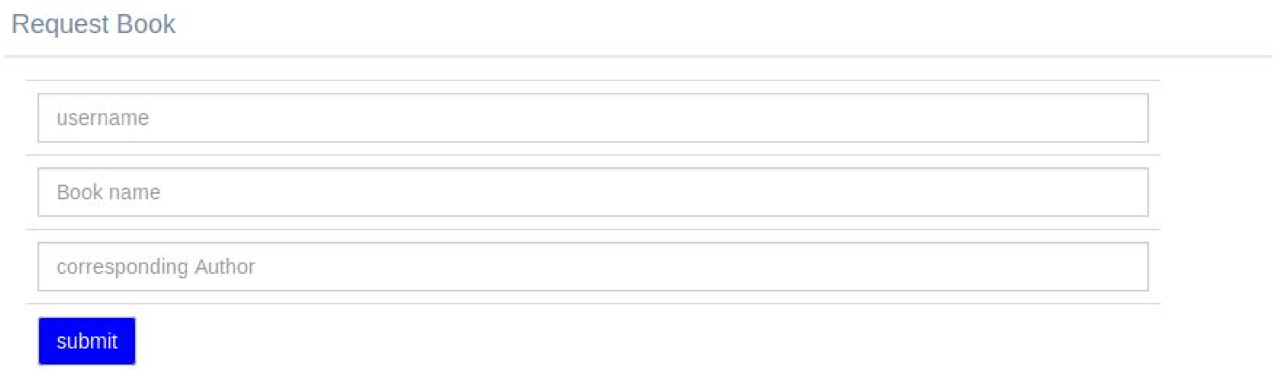
Return Information

ID	ISBN	Username	Issue Date	Fine	Paid
3	1234	aditya32	2017-05-06	300	N

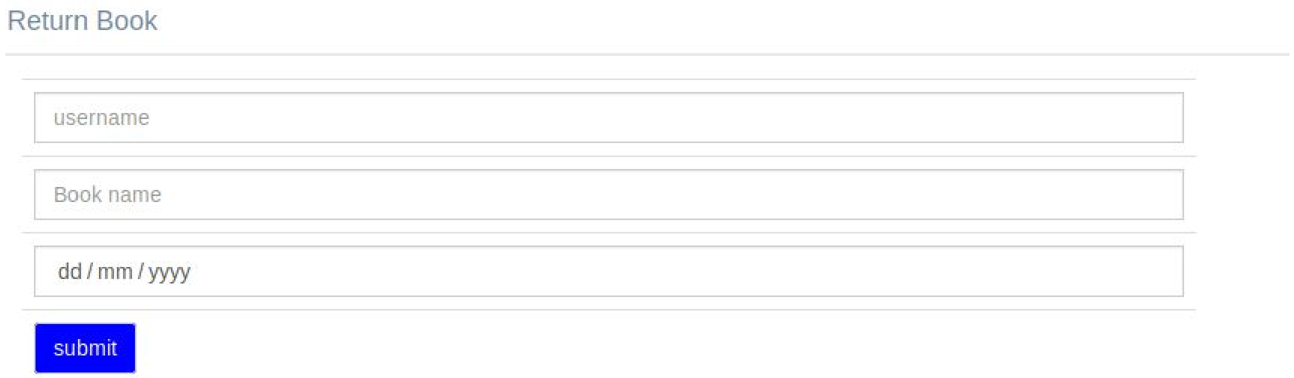
3. User Interface for ISSUE book :



4. User interface for requesting new books :



5. User interface for returning a book :



CONTRIBUTIONS :

It is not like one has done more and one has done less But we can say with full confidence that we have completed the project together with pure team work. For every part of the assignment we have discussed together and shared our ideas with each other and worked together the complete the project. If one want us to specify the contribution of each member we can say that :

K. DURGA PRASAD REDDY -

- Planned the Entity-Relationship Diagram together.
- Logical schema as per the Entity-Relationship Diagram together .
- Procedures, Roles and Privileges.
- Created test data and testing the procedures and triggers for various roles and checking their corresponding privileges and modifying some triggers and procedures accordingly.

M. ADITYA -

- Planned the Entity-Relationship Diagram together.
- Modifications for the logical schema such that it is in 3NF form together.
- Triggers and testing .
- Front-end part using (HTML,CSS,PHP)

REFERENCES :

- Database Concepts by Henry F. Korth, S. Sudarshan
- Database Management Systems by Raghu Ramakrishnan and Johannes Gehrke
- mariaDB tutorial
- Class notes