

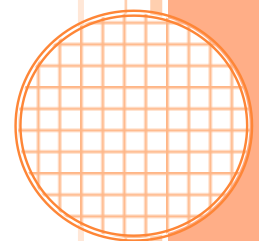
Project design report

# LIBRARY ENHANCEMENT:

*K.G. College of Engineering, University of Mumbai*

**Rucha Somani**

Abstract: The project focuses on enhancing the current library management system at K.G. College of Engineering. The librarians already have some system working for them, but the approach is cumbersome. They want to involve more technology so that management becomes easier.



# LIBRARY ENHANCEMENT: PROJECT SUMMARY

## Background:

Since a lot of books are issued and returned daily at the library, the system is majorly data dependent. The current solution that is being implemented by the K.G. College is partially online but mainly manual. The current system offers students an online portal to view the available books, but these entries are made manually by the librarian as and when he finds time. So, the system isn't very up to date as far as inventory records are concerned. Also, this system is very prone to manual errors and physical paper work is always in danger of being stolen or destroyed by fire, etc. The current system is also very labor-intensive.

The library at KG College, University of Mumbai, is a huge library with books from at the least, 12 engineering branches. The books in this library can only be borrowed by students and faculty at K.G College who have been issued a library card by the college. A person can only borrow two books at a time. Each book borrowed is due in 14 days. Returns made after due date are fined Rs. 3/day.

## Designed solution:

- Student/faculty (borrower) checks online which books are available and their location
- Borrower then goes to the library to borrow the book.
- The librarian scans the bar code sticker on the book which has unique number printed on it and inputs borrower's library card information.
- After the librarian inputs borrower's library card number, an authentication process runs to ensure that the borrower should be allowed to borrow a new book.
- This process will return affirmative only if the borrower hasn't exceeded 2 books and has paid all the dues applicable before he tries to borrow new book.
- Now the borrower can log in and see the book in his list of borrowed books and can also see the due date.
- On the time of return, the librarian scans the bar code again. The system calculates if any fine is applicable based on when the book was borrowed and the 14 day policy. If applicable, the system will show the librarian the amount of fine to be collected and the student pays fine.
- Administrator can Add/Modify/Delete user (librarian and borrower) accounts
- Librarians can add/modify/delete borrower accounts.
- Librarians can add a new book, modify existing information such as shelf number or publication, etc. or delete book.
- Librarians can see details of a particular borrower by input of their name or library card number.

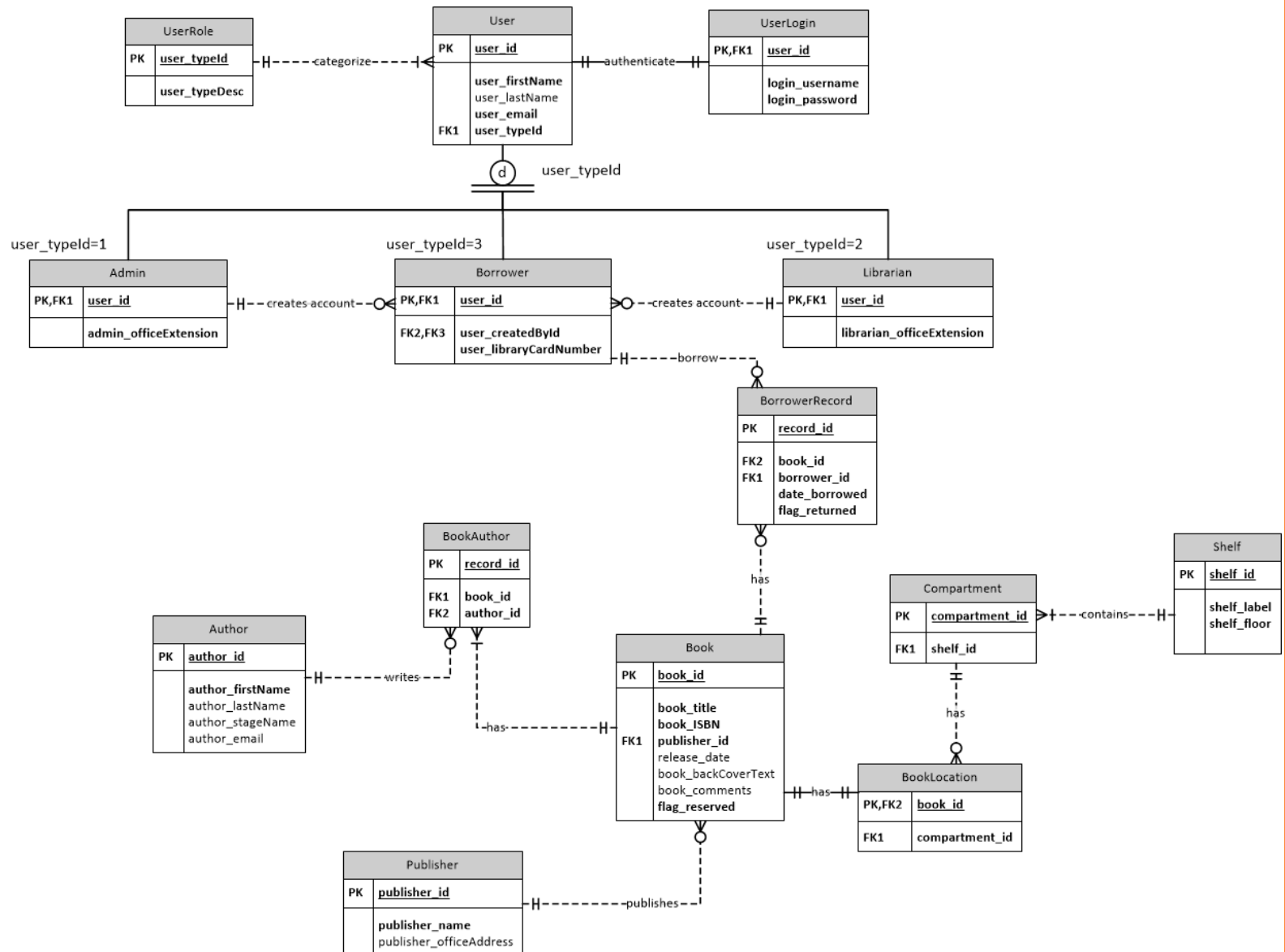
## TABLES AND ATTRIBUTES:

<b><u>DATAOBJECT:</u></b> <b>LMS_KGCollege</b>	<b>This database contains all the tables and relations that together build the Library management system for KG college.</b>
<b>Publisher</b>	<b>Stores all publishers records</b>
publisher_id publisher_name publisher_officeAddress	PRIMARY KEY: Each publisher will have a unique ID to identify publisher
<b>Book</b>	<b>Stores all the book details</b>
<u>book_id</u>  book_title book_ISBN publisher_id release_date book_backCoverText book_comments flag_reserved	PRIMARY KEY: Each book will have a unique ID to identify the book  FOREIGN KEY: Associated with primary key of 'Publisher' table  The descriptive text given on the back of the book  If the book is not supposed to be issued, flag is true.
<b>UserRoles</b>	<b>Stores the type of users. Reason for making a separate table is to provide future flexibility. New users roles can be added easily in future.</b>
<u>user_typeId</u> user_typeDesc	PRIMARY KEY: Each type of user is given a unique ID Contains each role of user. [admin, librarian or borrower]
<b>User</b>	<b>Contains all types of user information</b>
<u>user_id</u> user_firstName user_lastName user_email user_typeId	PRIMARY KEY: Each user has a unique user ID  FOREIGN KEY: Associated with Primary key of 'UserRoles' table. Determines the role. Possible values : 1,2 or 3
<b>Admin</b>	<b>Child entity of User. Contains admin information</b>
<u>user_id</u>  admin_officeExtension	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table.

<b>Librarian</b>	<b>Child entity of User. Contains librarian information</b>
<u>user_id</u> librarian_officeExtension	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table.
<b>Borrower</b>	<b>Child entity of User. Contains Borrower's information. Faculty of the college and students have the same privileges in the designed system, so distinction has not been made between the types of borrowers</b>
<u>user_id</u> user_createdById user_libraryCardNumber	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table. FOREIGN KEY: associated with primary key of 'Admin' table and primary key of 'Librarian' table
<b>BorrowerRecord</b>	<b>Maintains records of what books are borrowed by what users. Since many borrowers can borrow many books, this association table was created to avoid multivalued attributes.</b>
record_id book_id date_borrowed flag_returned	PRIMARY KEY: Auto incremented Integer identifying each record. FOREIGN KEY: associated with primary key of 'Book' table  Boolean. Will be false by default. True when the book is returned.
<b>UserLogin</b>	<b>Contains login credentials for users. Reason for making a separate table: Provides better security.</b>
<u>user_id</u>  login_username login_password	PRIMARY KEY: Each user has a unique user ID FOREIGN KEY: associated with primary key of 'User' table.
<b>Shelf</b>	<b>Contains information of shelves where books are kept</b>
<u>shelf_id</u> shelf_label shelf_floor	PRIMARY KEY: Each user has a unique shelf ID to identify name the shelf is known by floor where the shelf is located
<b>Compartment</b>	<b>Contains information of each compartment is a shelf</b>
<u>compartment_id</u> shelf_id	PRIMARY KEY: Integer identifying compartments of shelf FOREIGN KEY: associated with primary key of 'Shelf' table.
<b>BookLocation</b>	<b>Maintains records of where the books are located. It is created as a separate table because locations constantly get updated. A separate table would be more efficient.</b>
<u>book_id</u> compartment_id	PRIMARY KEY: Integer identifying each book individually FOREIGN KEY: associated with primary key of 'Book' table FOREIGN KEY: associated with primary key of 'compartment' table.

<b>Author</b>	<b>Maintains record of authors of the book</b>
<u>author_id</u> author_firstName author_lastName author_stageName author_email	PRIMARY KEY: Integer identifying each author uniquely  Some authors like to have a different name displayed on book
<b>BookAuthor</b>	<b>Maintains records of which books are written by what authors. Sometimes books can have multiple authors so this association table was created to avoid multivalued attributes</b>
<u>record_id</u> book_id author_id	PRIMARY KEY: Integer identifying each record. Auto incremented field. FOREIGN KEY: associated with primary key of 'Book' table. FOREIGN KEY: associated with primary key of 'Author' table.

## RELATIONAL DATA MODEL:



## BUSINESS RULES:

- Currently, the system has only 3 types of users. So user\_typeDesc of 'UserRole' table can take only 3 values, 'admin', 'librarian', 'borrower'
- Borrowers can borrow maximum 3 books.
- Each book borrowed must be returned in 14 days otherwise a charge of Rs 3 per day is fined.
- The "UserLogin" table is to be kept encrypted.
- Only the borrower accounts created by librarian can be deleted by librarian. Admin can delete all accounts.

## MAJOR DATA QUESTIONS:

Since the current system at KG College is a –part online-mostly manual system, they want to turn to a completely online system to manage their data efficiently.

The users using the system are of three types,

The admin  
The library staff  
And the borrowers

Following list shows segregation of what data questions each role needs answered by the proposed system and how we achieve it.

### **Why Borrower queries the database:**

- View which books are available

This list can be obtained by a join on 'Book' table and 'BorrowerRecord' table. The list will include all books in the 'Book' table for which there is either no record in 'BorrowerRecord' or has a record with 'flag\_returned' as true.

This includes searching a book using Author details or Publisher details. The database design also facilitates this by identifying each author and publisher and associating them to the book.

For eg. If a borrower is looking for books by author named 'Henry Korth', the program should locate the 'Author\_id' of this Author and then use that Id to locate records in the 'BookAuthor' table.

- Location of books in library

Location of books can be determined by selecting a book from 'Book' table and looking for that 'book\_id' in 'BookLocation' table. This gives the 'compartment\_id' where the book is kept. This 'compartment\_id' can be used to determine the Shelf. The Shelf table also contains the floor on which the shelf is placed. So thorough location can be obtained.

- View books currently borrowed by self and due date

This can be determined by selecting the particular user from 'User' table and using the 'user\_id' to locate records where that user\_id reflects in 'BorrowerRecord' table.

Due date as a business rule, would be programmed on the server side to reflect 14 days after the date mentioned in 'dateBorrowed' field of 'BorrowerRecord' table.

- View fine applicable of books borrowed by self that are past due date

After the records for books borrowed by self and due date are received, using a programmed function, this date will run through the code and any number of days past 14 days, after the book was borrowed, will be charged Rs. 3/- per day programmatically.

### **Why Librarian queries the database:**

- Add / Modify/ Delete Borrower accounts

Whenever the librarian adds a new borrower account or changes the current one or deletes an old one, the 'Borrower' table is updated. One restriction here is that a librarian can only Modify or Delete a Borrower Account that has been created by the librarian. To keep a track of this, the 'user\_createdById' field is used from 'Borrower' table. Whenever a new Borrower Account is created, this field stores which librarian created the account or [admin id, if created by admin].

If the librarian has the same id as the 'user\_createdById' of a borrower account that is being changed, then the librarian can proceed or else, there will be an error. This has to be implemented programmatically.

- Add / Modify/ Delete books of library

When an older book is deleted [theft, loss, etc.] the table that gets affected is 'Book' table. In case of deletion, the entire row represented by the 'book\_id' of the book, gets deleted. Where ever this 'book\_id' has child references, that rows must also get deleted.

When a new book is added, or existing book is modified, after filling up details in the 'Book' table, the corresponding 'book\_id' should also be updated in the 'BookLocation' table.

- View list of borrowed books

This list can be obtained by getting a list of 'book\_id's by looking at records from 'BorrowedRecords' table for which the 'flag\_returned' is false. This list of 'book\_id's can then be used to obtain other book details from the 'Book' table.

- Track a Book

To track a particular book, the 'book\_id' should be searched in the 'BorrowedRecords' table. This table will tell when the book was borrowed, by which user, whether it has been returned. To find other details of users who borrowed, the 'user\_id's should be run through the 'Borrowers' table.

### **Why Admin queries the database:**

- Add / Modify/ Delete Borrower accounts

Whenever the Admin adds a new borrower account or changes the current one or deletes an old one, the 'Borrower' table is updated. There is no restriction to the admin in terms of modifying or deleting accounts. Admin can delete any accounts created by admin or librarian. When a new Borrower Account is created, 'user\_createdById' field of 'Borrower' table, stores admin id.

- Add/ Modify/ Delete Librarian accounts

Whenever the Admin creates a new Librarian account or modifies the existing one, the 'Librarian' table gets updated. Deleting a librarian should ensure all the 'Borrower' Accounts created by the librarian being deleted are assigned admin id for the field 'user\_createdById'. This has to be achieved programmatically.