2025

LIBRARY BOOK SYSTEM

By The Alpha Crew

Table of Contents

PROJECT BACKGROUND

Problem

Libraries usually rely on the traditional way of doing things, the manual data capture. This method is prone to errors and data loss, may cause confusion and is time consuming. Means of backup for this method are also time consuming and costly. Managers and Librarians find it hard to keep track of activities happening within the library. Efficiency can also be compromised due to laziness of manually going through all the files to do reports and keep track of activities.

Solution

Development of a user-friendly Library Management System that only the Manager and Librarian are going to interact with. The system will enable the librarian to organize books; track loans and returns and manage member details quickly and accurately. The manager will also get to easily generate reports from this system.

Functionalities

Overall Functionalities

- **Authentication**: Staff members, including librarian and manager, can securely log in using a username and password.
- **Book Inventory**: The manager can add, view, edit, and remove books. This includes details like title, author, quantity, category, and availability status.
- **Student management**: create, read, update, and delete (CRUD) options are available for student records.
- **Borrow/Return**: Staff members can record and manage book loans. This includes issue date, due date, return date, and automatic updates on book availability.
- **Search**: Librarian or the manager can search for books by title, author, or category. They can also search for students by their email address.
- **Data validation & error handling**: The system ensures that required fields are filled out and checks input formats, such as dates.
- **Reporting**: The manager can generate reports such as overdue borrowed books, currently borrowed books by student and the list of books in the inventory.
- **Multiple forms**: Login Form, Main Menu Form, Book Form, Student Form, Borrow Form, Report Form are available.

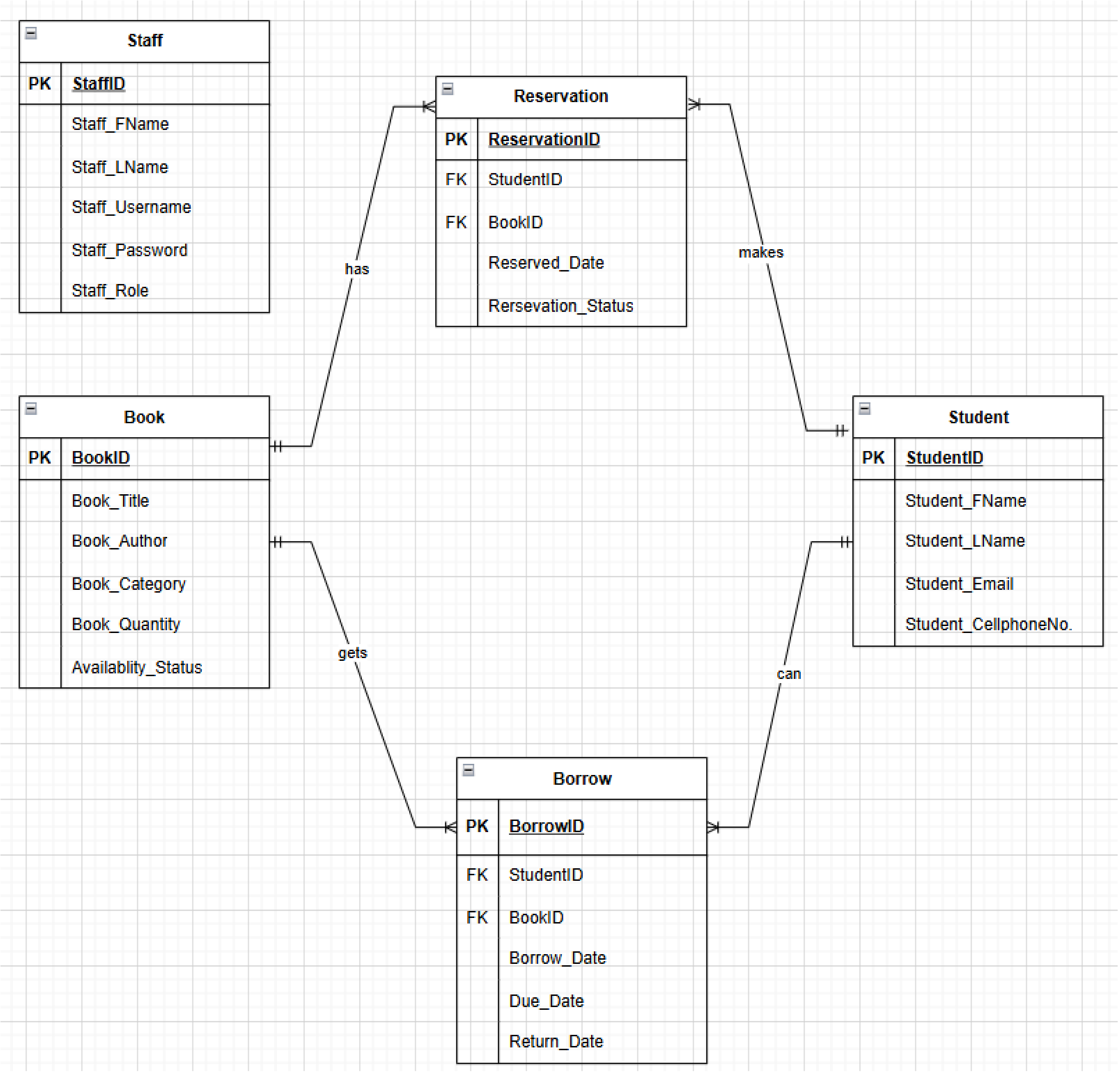
Manager Functionalities

- **View all database tables:** This function gives the overview of all tables to the manager.
- **Manage Library Staff**: This function gives the manager the privilege to add/modify/delete the library staff.
- **Add new book:** This allows the manager to add a new book entry to the book inventory, including details like author, title and quantity. Once added, the book becomes available for students to borrow based on the available quantity.
- **View overdue bookings**: Allows the manager to see a list of students who have not returned the books on time. The manager can use their contact details to follow up with them.
- **View Statistics:** This displays library usage analytics, such as most borrowed books, frequent borrowers and overdue trends. This helps identify areas for improvement and better decision making.

Librarian Functionalities

- **Add a new student:** This function registers a new student by collecting their email address, name, surname, and a cell phone number. The information is stored in the database and used to associate students with their future borrowing records.
- **View books offered by the library**: This function allows the librarian to see all the books available in the library, along with how many times each book has been booked. It also helps track the popularity and demand for specific books.
- **Issuing a book to a student:** This function helps the librarian to record and link a student to a book the student borrowed.
- **Reservations:**
  a) **View reservations**: This function allows the librarian to view all the reservations
  b) **Add a reservation**: This function allows the librarian to add a new reservation on behalf of the student.
  c) Helps manage queues for books that are currently fully borrowed and student who made the earliest reservation is notified when the book becomes available.
- **Returning a book:** This function helps the librarian to mark if a student has returned a book they had borrowed. It updates the system to reflect that the book is now available for others to borrow.
- **Borrow History**: This function helps the librarian to see the records of all books borrowed. It helps the librarian track borrowing patterns and generate reports when needed.

ENTITY RELATIONSHIP DIAGRAM



SQL QUERIES

a) SQL statement that extracts and displays all data from a database table

SELECT Book_title AS 'Title', Book_Author as 'Author', Book_Category AS 'Category', Book_Quantity AS 'Quantity' FROM Book

The query extracts and displays data from the Book table, and for better readability the column names were renamed. It allows library staff to view the books available in the system.

b) SQL statement that will update the values of a table

UPDATE Borrow SET Return_Date = @returnDate WHERE BorrowID = @borrowID

This query updates the Borrow table by recording the date the book is returned, this ensures accurate tracking of returned books. This is useful to the librarians to maintain accurate borrow records and prevent overdue dispute.

c) SQL statements with a WHERE clause

1. <u>Validate Staff Login</u>
   SELECT Staff_FName, Staff_LName, Staff_Username, Staff_Password, Staff_Role FROM Staff WHERE Staff_Password = @Password AND Staff_username = @Username
   This query validates a user's logins by checking their credentials against stored data. It is useful for ensuring that only authorised staff gain access to the system.

2. <u>Filter Books by Category</u>
   SELECT Book_Title as 'Title', Book_Author as 'Author', Book_Category as 'Category', Book_Quantity as 'Quantity' FROM Book WHERE Book_Category = @Category
   This query displays books that belong to a certain category chosen by the user. We get these books from the Book Table. This query would be useful for the librarian for searching books by subject area.

3. <u>Check if a Student Already Exists</u>
   SELECT COUNT (*) FROM Student WHERE Student_Email = @Email
   This query counts how many students share the same email from the Student Table; this helps to prevent duplicates and ensure data integrity.

d) SQL Subqueries

1. Book Borrow Validation

   SELECT (SELECT Count (*) FROM Borrow WHERE Borrow.StudentID = @StudentID AND Borrow.BookID = @BookID AND Return_Date is NULL GROUP BY Borrow.StudentID) AS ActiveBorrows,
   (SELECT Book_Quantity FROM Book WHERE BookID = @BookID) AS TotalQuantity, (SELECT Count (*) FROM Borrow WHERE Borrow.BookID = @BookID AND Return_date is NULL) as BorrowedCount

   This query uses multiple subqueries to validate book borrowing request. It performs the following checks:

   - ***Student's Active Borrow** -It checks whether the student already has an active borrow for the same book.*
   - ***Book Availability** – It checks the Book_Quantity column in the Books Table to see how many copies of the book are available.*
   - ***Books Currently Borrowed** – It verifies how many copies of that book are currently borrowed by other students.*

   These checks are important for the librarian because they help confirm that the book is available before approving the borrow request. This ensures that no more books are borrowed than what's available, students do not borrow the same book multiple times at once and all students have fair access to library resources.

2. Find Next Student for Reserved Book

   SELECT TOP (1) * FROM Student WHERE Student.StudentID = (SELECT TOP (1) Reservation.StudentID FROM Reservation WHERE Reservation.Reservation_Status = 'Pending' AND Reservation.BookID = (SELECT BookID FROM Book WHERE Book_Title = @Book_Title) ORDER BY Reservation.Reserved_Date ASC, Reservation.ReservationID ASC)

   This query finds the next student who has a pending reservation for a returned book, based on reservation date and order. It useful for notifying students when a reserved book becomes available.

e) Aggregation Queries

1. Count Number of Borrowed Books by Student

   SELECT StudentID, COUNT (*) AS TotalBorrowed FROM Borrow GROUP BY StudentID

   This query counts how many books each student has borrowed. It is useful to the librarian for monitoring borrowing patterns and high usage students.

2. Count Overdue Books

   SELECT COUNT (*) AS OverdueCount FROM Borrow WHERE Due_Date < GETDATE () AND Return_Date IS NULL

   This query counts the number of overdue books by checking the Borrow table. It compares the current date with the due date; it also checks that the return date is null meaning the book hasn't been returned.

f) Additional Queries Used in the Project

- Insert New Student

  ```
  INSERT INTO Student Student_FName, Student_LName, Student_Email,
  Student_CellPhoneNo) OUTPUT INSERTED.StudentID VALUES (@name,
  @surname, @email, @cellphone)
  ```

  This query adds a new student record to the database after validating email and phone number.

- View All Reservation Records

  ```
  SELECT Student.Student_FName + ' ' + Student.Student_LName AS 'Name
  and Surname', Student.Student_Email AS 'Email', Book.Book_Title AS 'Title',
  Reservation.Reserved_Date AS 'Reserved Date',
  Reservation.Reservation_Status AS 'Status' FROM Reservation JOIN Student
  ON Reservation.StudentID = Student.StudentID JOIN Book ON
  Reservation.BookID = Book.BookID
  ```

  It displays all reservation records in a user-friendly format. It joins three tables: Reservation, Student and Book, and shows the name and surname of the student, email address of the student, title of the reserved book, reservation date and status of the reservation.

GROUP INFORMATION

Name & Surname	Student number
Anele Ngubane	224056517
Aphelele Mtombeni	224031788
Mvelo Khumalo	224035943
Ndumiso Hlatshwayo	224046046
Nobuhle Ngwenya	224065281

Login Details

SQL Server Account Details

Username: ist2ko
Password: jioeox

Manager Details

Username: mvelo_k
Password: khum@789

Librarian Details

1. Username: anelen
   Password: ngub@123
2. Username: ndumih
   Password: hlats@012
3. Username: aphem
   Password: mtho@456
4. Username: buhlen
   Password: ngwe@345

ACKNOWLEGDEMENTS

1. AI (ChatGPT to be precise) was used to phrase the system background to be more professional with correct grammar. Moreover, it was used to debug certain methods responsible for event handling and functionality. Furthermore, it was used to generate the application icon.
2. The AI was also used to understand the concepts of database communication with the database and not relying on Visual Studio’s drag & drop feature. This includes learning how to use:

   -SqlConnection - which was used to establish the connection to database.

   -SqlCommand - used to execute SQL queries.

   -SqlDataAdapter - used populate datatables, especially when working with foreign keys.
3. The Icon and the background image used in the project were sourced from (https://www.pngwing.com/).

ISTN212 ASSIGNMENT PEER EVALUATION FORM

Group Name	
The Alpha Crew	
System Description	Library Book System used to manage and capture book and student details easing the labour of the library staff.

No	Group Member's Student Number	Group Member's Name & Surname	% Estimated Contribution to assignment	Signature
1	224035943	Mvelo Khumalo	20%	
2	224056517	Anele Ngubane	20%	
3	224031788	Aphelele Mthombeni	20%	
4	224046046	Ndumiso Hlatshwayo	20%	
5	224065281	Nobuhle Ngwenya	20%	

Total 100%

The estimated individual contribution column will reflect an intuitive assessment that is made holistically by the entire group for each group member. This assessment should be a consensus-based value where the entire group agrees on the amount of effort that individual group members have made in order to ensure the success of the assignment.

In most cases, where the group has worked as a well organised team, this assessment value will reflect equal values for all group members. However, in cases of exception where some group members may have contributed more/less than other group members, then this should be appropriately reflected. The percentage sum of the individual contributions must add up to 100.