

Gavin White

Databases

15/1/2024

# Implementation and User Manual

<b>Introduction:</b>	<b>2</b>
<b>Useful Links:</b>	<b>2</b>
<b>Process:</b>	<b>2</b>
<b>User Manual:</b>	<b>5</b>
<b>Password table:</b>	<b>6</b>
<b>Customer table:</b>	<b>6</b>
<b>Index Page:</b>	<b>6</b>
Manager View:	7
<b>Add New Customer Page:</b>	<b>7</b>
<b>Book List Page:</b>	<b>7</b>
<b>Edit Page:</b>	<b>8</b>
<b>Book Details Page:</b>	<b>8</b>
<b>Checkout Page:</b>	<b>8</b>
Customer View:	9
<b>Book List (Customer):</b>	<b>9</b>
<b>Register As New Customer(Customer):</b>	<b>9</b>

## Introduction:

In this report, I will go over the process of the Library Inventory Management System I created in Visual Studio and have uploaded to the ysjs.net server.

## Useful Links:

[https://ysjs.net/~gavin.white/Databases\\_Assesment/](https://ysjs.net/~gavin.white/Databases_Assesment/) - Link to LMS on ysjs server

[git@git.ysjs.net:gavin.white/databases\\_assesment.git](git@git.ysjs.net:gavin.white/databases_assesment.git) -SSH gitlab link to code

[https://git.ysjs.net:8888/gavin.white/databases\\_assesment.git](https://git.ysjs.net:8888/gavin.white/databases_assesment.git) -HTTP gitlab link to code

<https://screenrec.com/share/8EW5tKxkbQ> -Video Link

## Process:

The first step I took in this project was creating the database in ysjs MySQL server. This was pretty simple as I had already done the specification. The next step I took in implementing this database system was creating the index/login page. I wanted to create a way for the manager to login that was password protected. I originally wanted to create a hash function to store the password securely but ultimately, I decided that was out of the scope of this project. I also wanted to add a way for a customer to go to their part of the website that would allow them to register as a customer, so I added a button on the index. The next page I created was the customer page, which allows them to register as a new customer. I did this using php to query the database, check if they already exist, then add them as a new customer if they don't. Next, I created the book list page for the customer to view and search all of the books in the database. For this, I created a table that uses queries of the database to fill out the table. This took a lot of

effort and time as I wanted a table to display the author's name and not just their ID. Due to the author's name being in a separate table to the rest of the information, I had to use a very complex query. Because I had already created this book list table, the next thing I added was the book list table that the manager would use which meant adding a view details, edit, and delete button into the table on each row. I also added a button that allowed the manager to add a new book to the database. I then went on to create the add new book page and form. This turned out to be pretty simple and used fairly simple queries. Adding error messages did turn out to be difficult for me simply because I couldn't figure out the formatting for a while. I then created the page to which the details button in the book list populates, which was very simple. I decided to use the url based on the book clicked to make sure only the details for that specific book were displayed. The next two pages were created with the same process with the edit and delete pages and forms being made. The only hiccup I had developing these pages was making a cascade delete on book copies. When a book was deleted, I did not have the rights to create triggers in the ysJ MySQL server. I got around this by creating a query that deleted all the copies with the books matching ISBN number as well as deleting the book itself. I had a similar issue with the edit form but used a similar method to get around the lack of triggers. The final page and form I created was the checkout form, which caused me the most trouble. I used the book list table I created earlier to make it easy for the manager to check the book ISBN and any details he might need. Similarly, I added a table that showed the checkouts and the information that would go along with it. Finally, I added the checkout and return forms, which required the most complex queries yet because I wanted to make sure the system checked if there were copies available to checkout. I incorrectly assumed before I implemented these forms that this would be the easy part but it turned out to

be very wrong as this is the page and query I spent the most time debugging. The last couple things I did to finish up this database was add a way to prevent SQL injection by escaping the values in every form. Lastly, I added a nav bar to make it easy to navigate, and I made sure the customer could only access the pages they were allowed to view.

# User Manual

**Password table:**

	Fname	Lname	Username	Password
Manager 1	Martin	Tyler	M.tyler	manager1
Manager 2	Tyler	Long	T.long	manager2

**Customer table:**

	Customer_ID	FullName	Email	Street	Registration_date	City
Customer 1	1	Marissa Link	marissa.link@example.com	123 Forfun St.	2024-01-18	York, UK
Customer 2	2	Brad Pitt	brad.pitt@hollywood.com	456 Hollywood Dr.	2024-01-18	Los Angeles, USA
Customer 3	3	Walker Hayes	walker.hayes@example.com	589 Vegas Blvd.	2024-01-19	Los Vegas, USA

**Index Page:**

There are two different ways to use this system and page. One is as a customer and the other is as a manager. If you are a customer, you should click the “I am a customer” button in the top right corner of the page. This will bring you to the list of books within the library. If you are a manager, you should use one of the usernames and passwords as provided and enter them into

their respective fields. Once the correct credentials are inputted and the login button is pressed, you will be taken to the “Add New Customer” page.

### **Manager View:**

#### **Add New Customer Page:**

Adding a new customer is fairly simple. Simply input information into the provided fields and press the register button. There will be a check for a valid email so make sure the email provided could be valid. (As long as an “@” and “.com” are there, it will accept it so [username@example.com](mailto:username@example.com) would be valid.) The back page takes you to the manager's version of the book list, and the nav bar at the top links to the respective pages.

#### **Book List Page:**

This page and form allow a manager to perform all of the CRUD functions. If you would like to add a new book, click the “Add a new book” button in the top right corner of the page, this will take you to the add new book page and form. You can use the search bar above the table to search for any book you want, using Name, Author, or ISBN. You only need one of those fields to find the book you want. Once you have chosen a book, the details button will take you to the “Details” page. The edit button will take you to the page and form to edit that book. And the delete button will delete the book, and take you to a page that confirms the deletion of the book and all of its copies.

**Edit Page:**

The edit page is fairly simple. The fields will automatically be filled with the information of the book you selected. You can then edit any of the fields; press the edit button and you will get a confirmation or error message. The back button takes you back to the manager's book list.

**Book Details Page:**

This page is very simple. It displays the selected books information in a slightly easier to read format and also shows the rental price, which is not shown in the table.

**Checkout Page:**

This page and form allows you to both checkout and return a book. If you already know the ISBN and customer ID you can simply type them into the fields above the checkout button. Once the button is clicked, you will get a confirmation message if the checkout was performed successfully. You can also use the search table to find the book and ISBN you would like. If you do not know the customer ID, you can reference the table above of customers to make a checkout. Once a checkout is made, it will show up in the checkout table. You can then reference that table to enter the ISBN and customer ID into the fields above the return button to make a return. You will have to reload the page to see any updates to the table be reflected in the table.

**Customer View:**



**Book List (Customer):**

This page allows a customer to search the system for any book based on its ISBN, Name, or Author. Notice that it does not have a details, edit, or delete button. You must be logged in as a manager to see that page.

**Register As New Customer(Customer):**

This page works the exact same as the manager's version, but is available to the customer to do as well. Simply input your information into the required fields and press the register button.