**Senior Project: Book Rental Web System**

David Ubah, Justin Cox

Spring 2017

Table of Contents Page

Topic1

1.1 Problems 1

1.2 Description1

1.3 Scope2

Team & Task 2

Project Breakdown3

3.1 Architecture 4

3.2 Supporting Images5

3.3 Data Analysis9

Challenges 10

Future10

Additional Notes 10

References11

1. ***Topic:***

This is a web system that allows multiple users from any location to search for an available book, rent an available book, view the unavailable books, update/add new books, and view details of a rented book for a single library location.

***ACM classification keywords: Information system, Data management system, Query languages***

* 1. ***Problem:***

This project is specifically designed for librarians to implement a rental system that takes information about the book(s) being rented, the customer, and late fees then organizes them for both the librarian and customer. The project will keep track of book inventory and late fees efficiently so the librarians can focus on other responsibilities.

***1.2 Description:***

This is a web system that allows both users and administrators search for book availability, rent an available book, (admin) add new books, and view details such as author information and general description of the book. The project will be broken into several sprints, and each sprint will have its own separate goal with deadlines. Since there are only two members on the team, the tasks will be divided in half but both members will be expected to assist the other if help is needed. The only required resources we will need are a simulated server (XAMP), a code editor (Sublime text), and a database system (MySQL Maria DB).

***1.3 Scope:***

This project was broken in three sprints (developmental phases), which include:

A. Sprint 1:

* Creating/wire framing a user- interface design for the project
* Creating a system use case for the project

B. Sprint 2:

* Development of the project using the use case diagram.
* Creating a database for storing the user’s information, rental information and book information.

C. Sprint 3:

* Embedding security measures such as the use of regular expressions and stored procedures into the system for data validation.

***2. Team & Task:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Members** | | | |
| **Names** | **Skills** | **Roles** | **Responsibilities** |
| David Ubah | PHP, Html, CSS, JavaScript. | Front End Developers | * Design each page layout * Determine what data needs to be captured * Design textboxes and radio buttons * HTML coding |
| Justin Cox | MySQL, PHP, Java, JavaScript, Html. | Back End Developers | * Designing and reviewing database procedures * Determine table relations * Database security (SQL Injections) |

***3. Project Breakdown:***

There are two types of users who will use the Library system. This includes the book renter (user) and the librarian (admin). The book renter can register, login, search, rent a book and view the details of the rented book.  The librarian can do everything that the book user can do with the addition of updating books, and viewing unavailable books.

1. Registration use case enables users who are not already in the database as renters and librarians create accounts by providing their information such as name, email, password, and username.
2. The login use case enables users to access their account using their username and password for authentication and authorization.
3. The search use case enables the user to search for available books in the library.
4. The rent use case enables users to reserve a book that is available for a specific time period. This also allows the system to collect payment information for late return fees.
5. The update use case enables the librarian to add new books to the database to displaying the available books of the library with its description and availability and also enables the librarian to ensure the return of a book.
6. The view details use case enables the librarian and user to view the details of a rented book to identify lateness.

Actors:

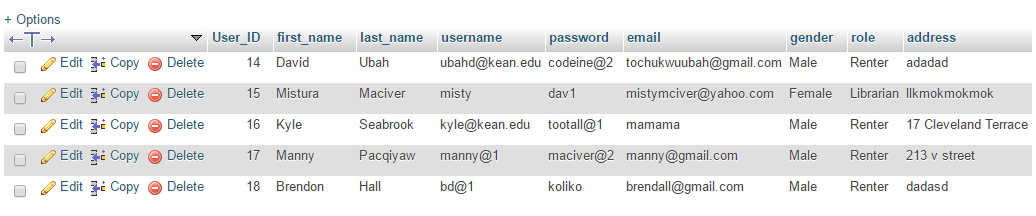
* Renter (User)
* Librarian (Administrator)

***3.1 Architecture:***

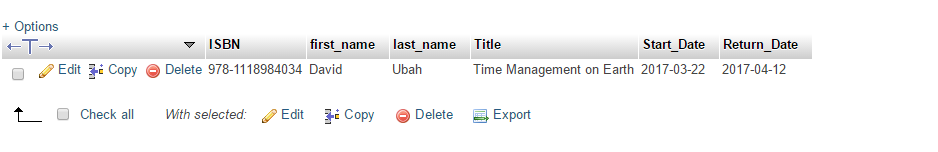
For this project, the following languages and framework were utilized:

1. XAMPP: gives the ability to develop locally and also provides a local database management system.
2. PHP: a general-purpose scripting language that is used for web development.
3. CSS: used for styling webpages.
4. Adobe Illustrator: used for creating page designs and logos.
5. MYSQL: database management system.

***3.2 Supporting Images***

******

***Figure 1. User table for storing the information of the users.***

******

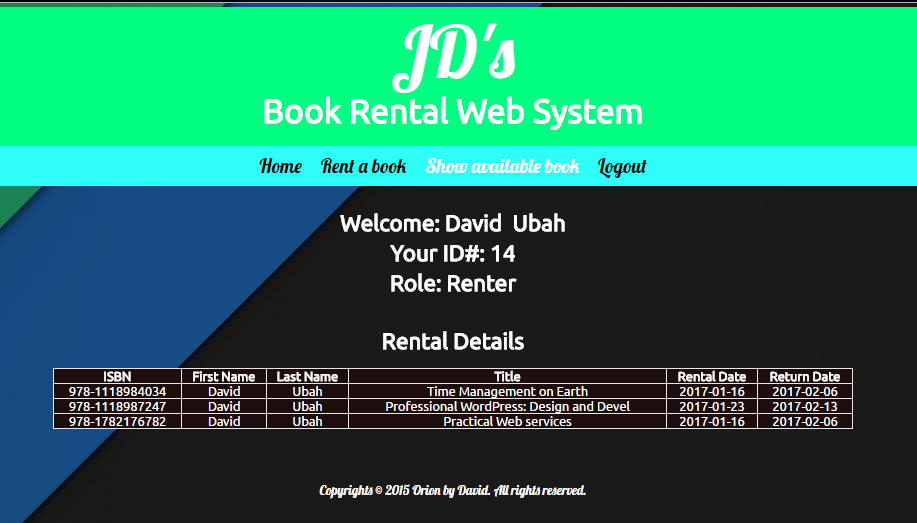
***Figure 2. Rent table for storing the renter’s information.***

******

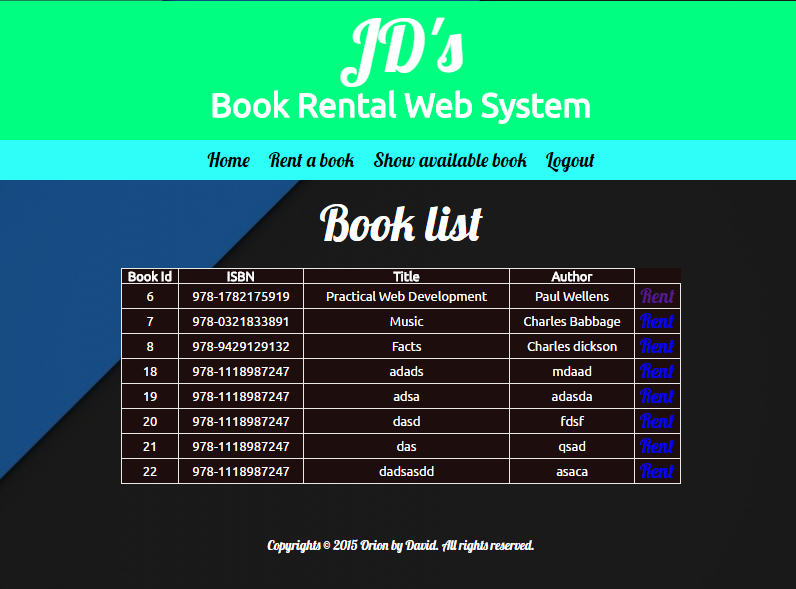
***Figure 3. Book table for storing the book information.***

******

***Figure 4. System homepage before login.***

******

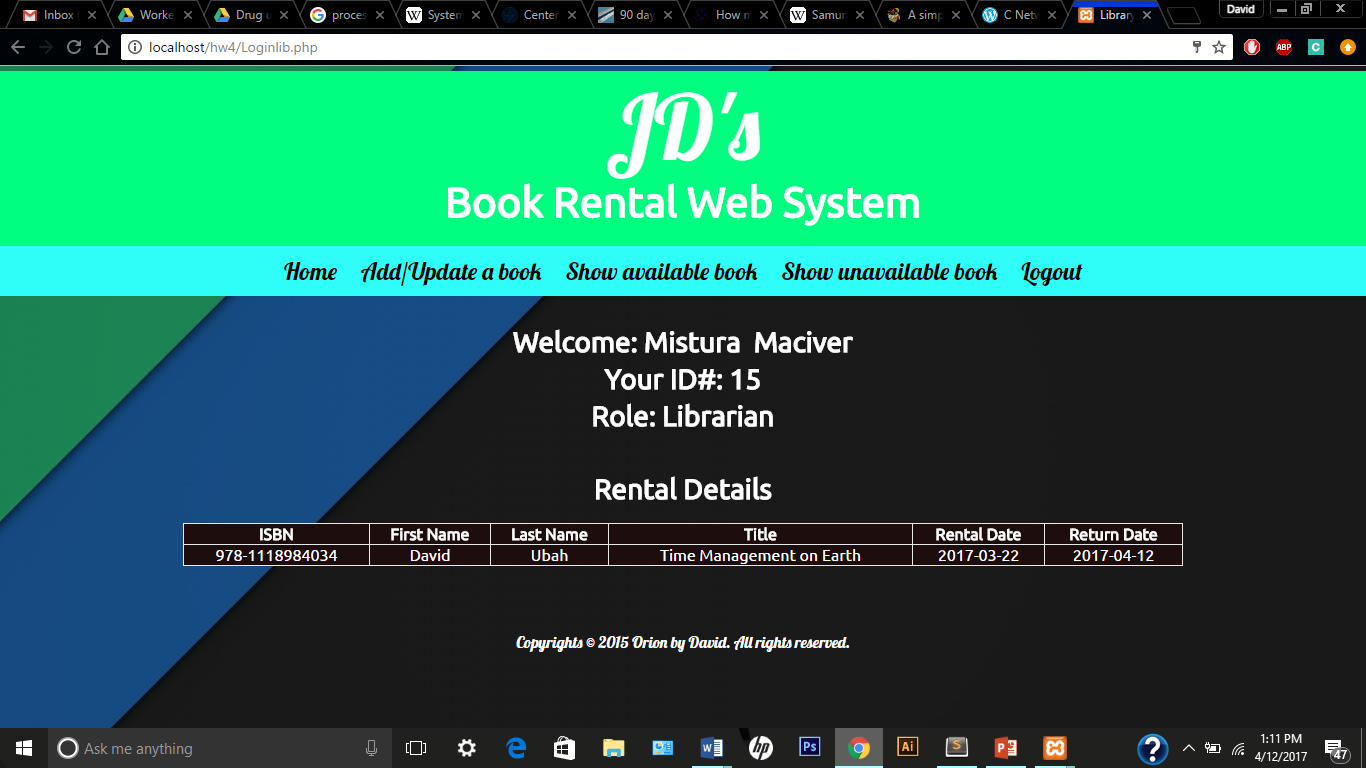
***Figure 5. Renter’s homepage after login***

******

***Figure 6. List of available book with its details***



***Figure 7. Registration page***



***Figure 8. Librarian homepage after login***

***3.3. Data Analysis***

Search available books

Update/Add books

<<includes>>

Save information to database

Input book description

Book Renter

Login

Librarian

Book Rental Web System

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<includes>>

<<includes>>

<<includes>>

<<includes>>

<<includes>>

<<includes>>

Register

Input user information

Input dates available

Input book information

Search for books

Purchase by credit card

Buy book

Select book

Check in book

Authorize as renter

Authorize as administrator access

Authorize access

Authenticate by password

Authenticate by username

Authenticate ID

***4. Challenges:***

The following challenges were faced during the development of the project:

1. Data management: When creating the database we had to determine which tables belong together, which tables could share values, and to make sure database doesn’t crash when presented with a large load.
2. It’s always difficult determining which languages to use, considering the advantages and drawbacks of each.

***5. Future:***

For the future, we hope to improve upon the design of the system in order to make it a bit flashier in design and page structure. Another improvement might be expanding the system into a distributed system and being able to work for larger scale systems that include other types of media.

***6. Information System Security Measure:***

For this web system, we will be making use of regular expressions to support input validations that will serve to protect the system and vital database information from SQL injections or attacks.

***References***

1. “A Quick-Start Tutorial on Relational Database Design,” *Relational Database Design*, 02-Sep-2010. [Online]. Available: https://www.ntu.edu.sg/home/ehchua/programming/sql/relational\_database\_design.html. [Accessed: 05-Apr-2017].
2. “HTML Layouts,” *HTML Layouts*, 20-Sep-2012. [Online]. Available: https://www.w3schools.com/html/html\_layout.asp. [Accessed: 05-Apr-2017].
3. “Oracle Database Security,” *Oracle Database Security*, 15-Jan-2015. [Online]. Available: https://www.oracle.com/database/security/index.html. [Accessed: 05-Apr-2017].