International
Burch
University

CEN 308 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

“M-D” library management system

Prepared by:
Dinko Omeragić
Mirza Nikšić

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

19.06.2021.

Table of Contents

1. Introduction

1.1. About the Project

“M-D” library management system is an automation solution for a library management. It is a web application used to keep track of all the books, users, employees and borrows inside a library. Librarians are able to register users, keep track of borrowed/available books and borrow details, while managers can supervise all transactions and keep workers in check. Link to where the application is deployed is the following: https://m-d-library-management-system.herokuapp.com/. But, we encourage you to go through the README file in our GitHub repository before using it. Link for the GitHub repository is the following: https://github.com/omeragicdinko/M-D-library-management-system.

1.2. Project Functionalities and Screenshots

Since all of the features are self-explanatory, we will only list them in the following paragraph.

List of main features:

- Employee and customer registration
- Updating customer and employee information
- Deactivating customer profile
- Creating books
- Updating books
- Deleting books
- Borrowing books
- Employee and administrator login
- Book search
- Customer search
- Employee search
- 0 books with the “Book Name” left notifier (manager is notifier when the last book with a certain name is borrowed)
- 1 book with the “Book Name” left notifier (manager is notifier when the last book with a certain name is left or available again)

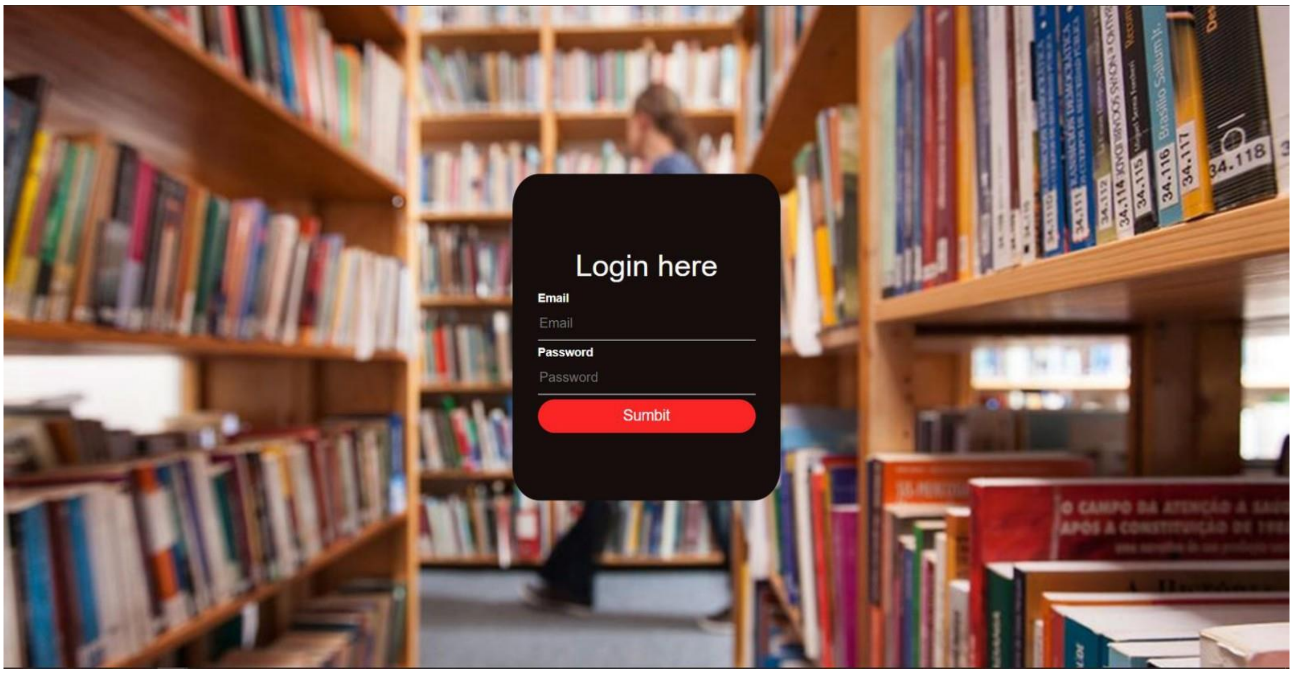
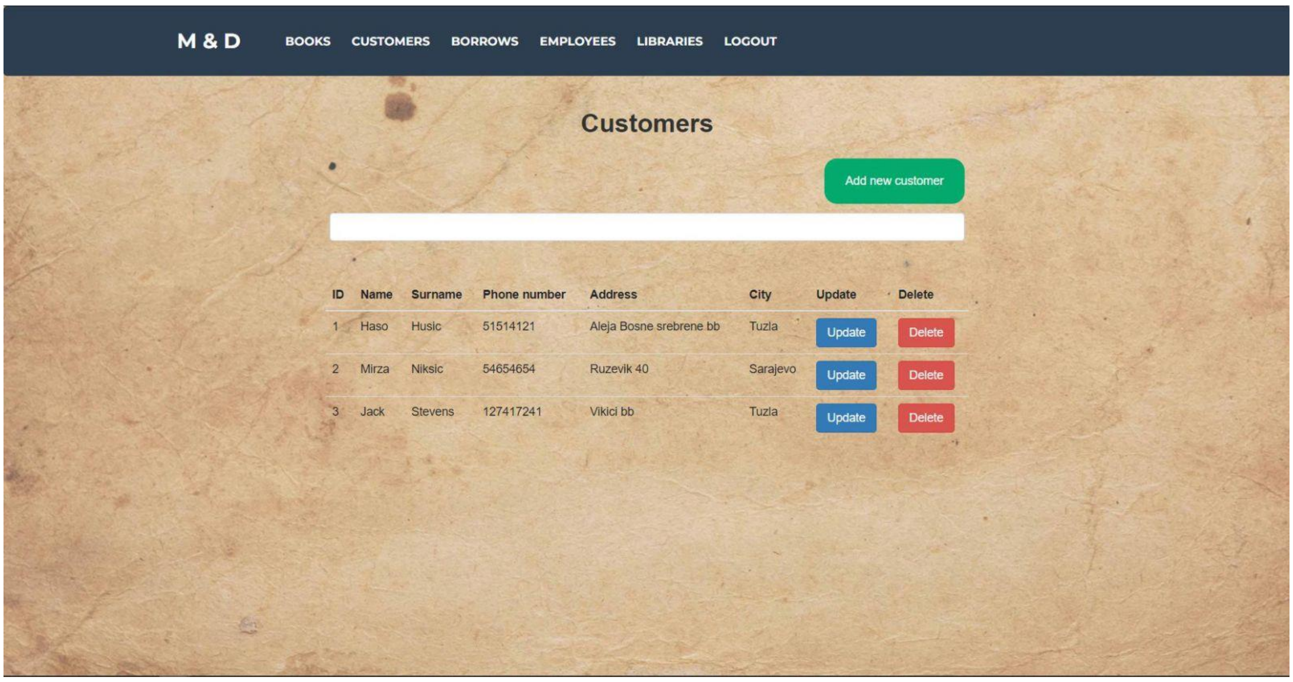
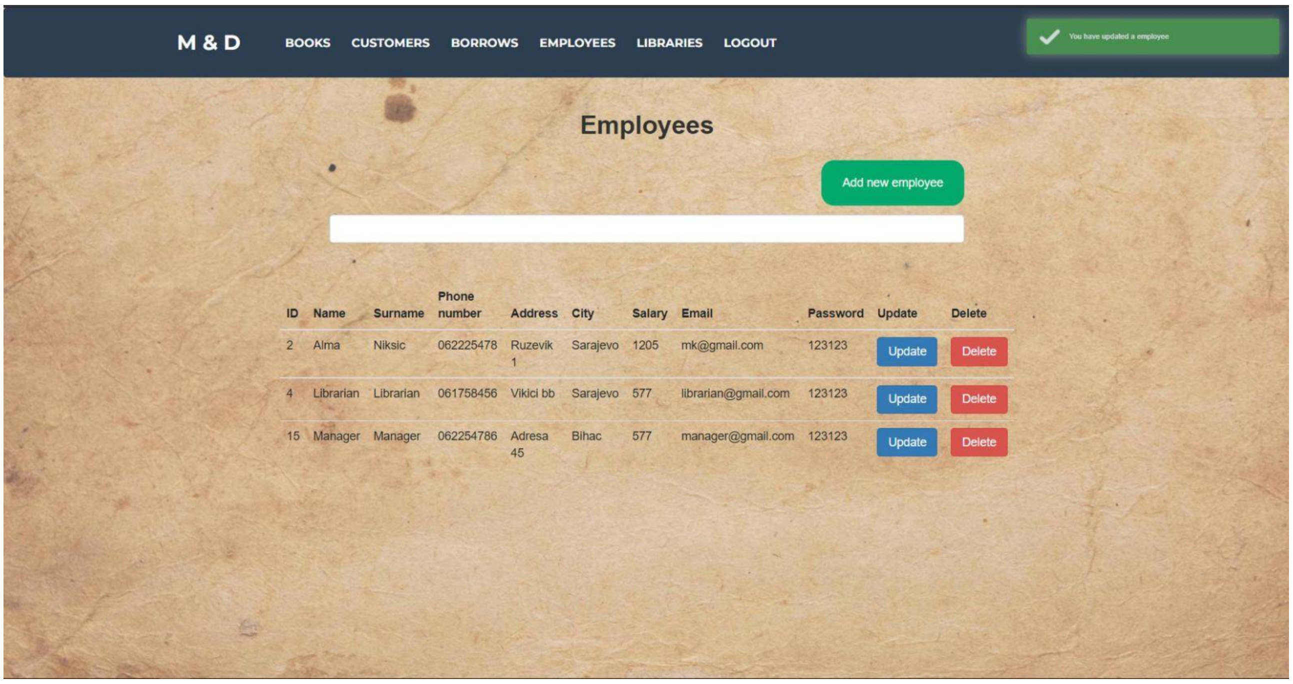


Figure 1: Login Page



ID	Name	Surname	Phone number	Address	City	Update	Delete
1	Haso	Husic	51514121	Aleja Bosne srebrene bb	Tuzla	Update	Delete
2	Mirza	Niksic	54654654	Ruzevik 40	Sarajevo	Update	Delete
3	Jack	Stevens	127417241	Vikici bb	Tuzla	Update	Delete

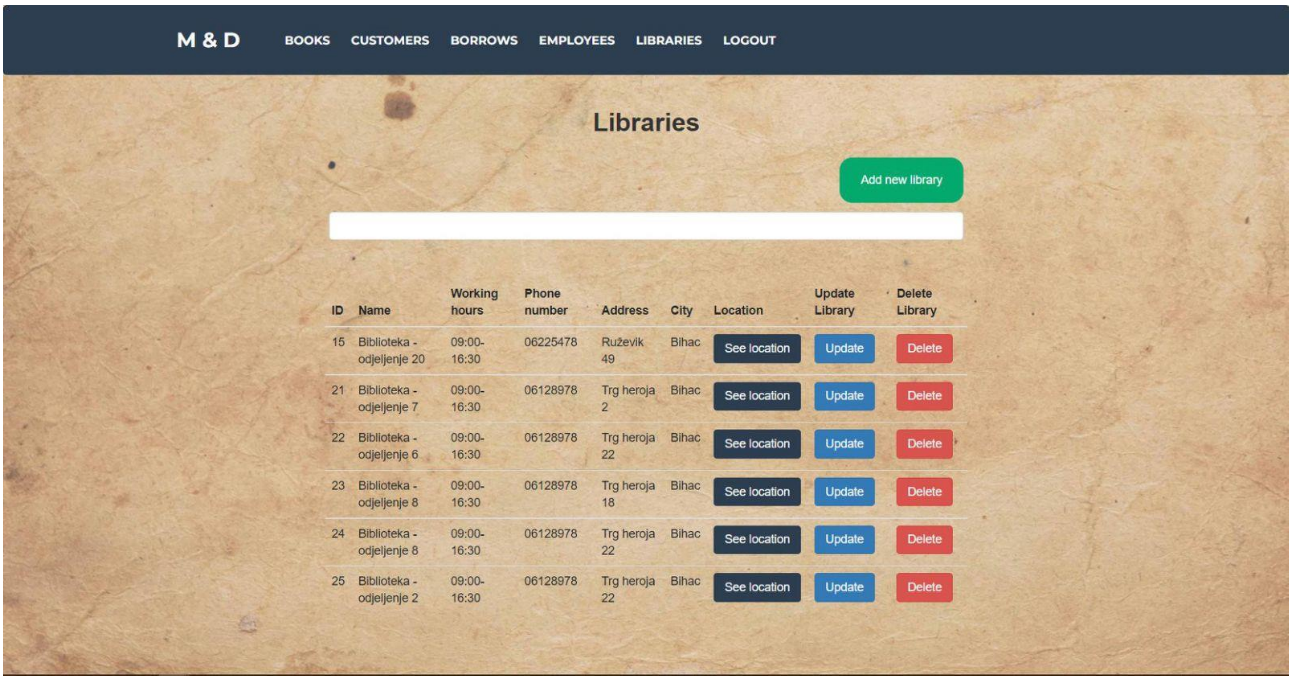
Figure 2: Customers Page



Employees

ID	Name	Surname	Phone number	Address	City	Salary	Email	Password	Update	Delete
2	Alma	Niksic	062225478	Ruzevik 1	Sarajevo	1205	mk@gmail.com	123123	Update	Delete
4	Librarian	Librarian	061758456	Vikici bb	Sarajevo	577	librarian@gmail.com	123123	Update	Delete
15	Manager	Manager	062254786	Adresa 45	Bihac	577	manager@gmail.com	123123	Update	Delete

Figure 3: Employees Page



Libraries

ID	Name	Working hours	Phone number	Address	City	Location	Update Library	Delete Library
15	Biblioteka - odjeljenje 20	09:00-16:30	06225478	Ruževik 49	Bihac	See location	Update	Delete
21	Biblioteka - odjeljenje 7	09:00-16:30	06128978	Trg heroja 2	Bihac	See location	Update	Delete
22	Biblioteka - odjeljenje 6	09:00-16:30	06128978	Trg heroja 22	Bihac	See location	Update	Delete
23	Biblioteka - odjeljenje 8	09:00-16:30	06128978	Trg heroja 18	Bihac	See location	Update	Delete
24	Biblioteka - odjeljenje 8	09:00-16:30	06128978	Trg heroja 22	Bihac	See location	Update	Delete
25	Biblioteka - odjeljenje 2	09:00-16:30	06128978	Trg heroja 22	Bihac	See location	Update	Delete

Figure 4: Libraries Page

2. Project Structure

2.1. Technologies

Technologies which we have used in the project are:

- HTML & CSS and JS(jQuery) - frontend
- PHP and FlightPHP – backend
- SQL – database

For PHP we have used PSR-12 coding standard. As for the others, we have structured the code according to the best practices in each of them.

We have also used two third-party libraries, they are: PHP-JWT and SwiftMailer.

2.2. Database Entities

Table names in our database are:

- books
- borrows
- customers
- employees
- libraries

2.3. Architectural Pattern

Architectural patter which we have used in this project is called the Layered architecture pattern. It is a pattern which organizes the system into layers with related functionality associated with each layer. A layer provides services to the layer above it so the lowest level layers represent core services that are likely to be used throughout the system.

We wanted to use the advantage of its flexibility because this project might scale in the future and having the opportunity to replace entire layers so long as the interface is maintained is a great advantage to have.

2.4. Design Patterns

Design patter which we have used in this project are:

- DAO (Data Access Objects) patter: used in backend, in all files in the folder *rest/dao*
- Observer pattern: used in backend, in the following files:
    - *rest/Subject.php*
    - *rest/Mail.php*
    - *rest/NumberOfBooksLeftNotifier.php*
    - *rest/index.php* (line numbers 142-164)

2.4.1. DAO (Data Access Object) pattern

Since we are mostly dealing with tables and CRUD operations on the data, writing all code inside the index.php file would make it messy. Therefore, we decided to use the DAO pattern to isolate the application/business layer from the persistence layer using a relational database. This makes it possible to see what is happening in each layer separately and makes the bug detection easier and faster.

2.4.2. Observer pattern

Since it is of great importance to keep track of available and the number of available books in the system, we decided to make a notifier for the manager to know if there is 0 or 1 book with the same name left in the library. To resolve to issue we have implemented the Observer pattern in which the manager would be the observer and the update will send emails to him/her in case this happens. This makes is easier for the manager to keep track of these occurrences without having to go through the tables and count each book.

3. Conclusion

To conclude, we are satisfied with the overall project implementation which we managed to do. The system is ready to be used in a real environment and it should provide the basic functionalities which a library management system should have. As for the future, we would modify the user interface after receiving feedback from real users according to their observations.

The most challenging part has been to find a solution for the “Tracking the number of books left” problem, but after visiting the Refactoring Guru website we have found a design patter which we used later on to resolve this issue.