

# Technical Manual

Jaume Pons Morillas

2n DAW

Library Project

# Introduction

This manual will explain how the web application is working. To do so it will explain how the different interactions that the client can do with the server are managed.

## Login

From the homepage or the catalog page a registered user can login into the website. When the user fills the login form the data is sent to the server via POST. The information is recovered in the `utilities.php` that will call the login function. The login function will check if the user credentials (now hardcoded) are correct, if they are correct a new SESSION will be started and a SESSION cookie will be automatically generated, the system will also store on SESSION the user name, user id and user type. The user will now have access to all the functionalities that it's permissions allow.

## Logout

When a user clicks the logout button the systems sends by GET data to `utilities.php` that will call the logout function. This function will unset the session, destroy the session and delete the cookie from the user browser.

## Dynamic menu

To achieve a dynamic menu the system will check if a user is logged in and retrieve the user type from SESSION. Different options on the menu will be loaded depending on the user type.

## Search function

The search function is implemented on the catalog page that will act as a results page. When a user does a search from the homepage information will be sent to the catalogue page via GET. The catalog page will process this info and call the `searchBooks` function from `utilities.php`. This function will search on the repository all the books that match the search parameters and then it will call the function `loadBooks` that will add all the necessary code to the catalog page to display all the results.

## Display elements

To display elements the systems uses the `loadbooks` function from `utilities.php`. This function can work in two ways:

1. If only one argument is passed (a number) the function will return the code to display that number of elements
2. If the first argument is 0 and the second argument is an array with elements ids the function will retrieve the elements using the searchById function and will generate the code to display them.

## Displaying user data

To display user data the system calls the findUser function from utilities.php. This function will retrieve the user object with all the data.

## Utilities.php

The utilities class is a bit special and needs some detailed explanation. As at the time of doing the library I didn't know how to use databases from php all the information is hardcoded. The place where all this information is stored is on utilities.php.

When I found that a php function cannot access a variable on the same file the only workaround that I found is to declare the class as abstract and declare some static attributes.

When databases are implemented utilities will be changed to a structured programming file and all the info will be on the database.

To initialize all the data utilities has a init function that creates all the needed objects and stores them on the static attributes.

To access those attributes is always done from functions as when databases are implemented I will only need to change the code inside that functions.