

The Technical Library

The technical
manual for The
Library Project



Index

The Technical Library	1
The project, The Library	3
Requirements	3
Software	3
Hardware	3
Database	4
Deployment	4
Diagrams and design	5
Data dictionary	5
Book table	5
Copy table	5
Lend table	6
Reserve Table	6
Users Table	6
Classes diagram	7
Available	7
Copy	7
MakeForm	7
MakeTable	8
Lend	8
Book	9
Reserve	10
ShowData	10
User	11
Source code	12
Structure	12
Known bugs	12
Future improvements	13
Useful information	13
Comments	13



The project, The Library

This project is a prototype of a website to manage a library. This manual was created to show how it is working The Library inside. Please feel free to modify, share, or ask for anything to my e-mail daw2ababotllinares@iesjoanramis.org.

Keep in mind that this is my first "big" web project. That does not mean that you can not criticize or suggest changes, please do it.

This project is **Creative Commons 4.0 International**.

This means you are free to:

- **Share:** copy and redistribute the material in any medium or format
- **Adapt:** remix, transform, and build upon the material for any purpose, even commercially.

Under the following terms:

- **Attribution:** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike:** If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions:** You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Hope you enjoy the project. Free culture!

Requirements

Software

The Library is developed basically in PHP and HTML5. For the style is using CSS3 and Bootstrap libraries. In addition, there are some functionalities developed in JavaScript. For the Data Base, we use SQL language in MySQL.

At the server, the only requirement is to have MySQL installed. At the client side it's important to have cookies and JavaScript enabled. Is good if the client uses the latest versions of the browsers. The Library doesn't support Internet Explorer in a version less than 10.

Hardware

There are not hardware Requirements for the server, just depend on the demand you will have. This needs a specific study of your project.

For the clients who will be used by librarians normally, it would be nice if is a bar code reader, this will make the work easy and fast, so we recommend it.

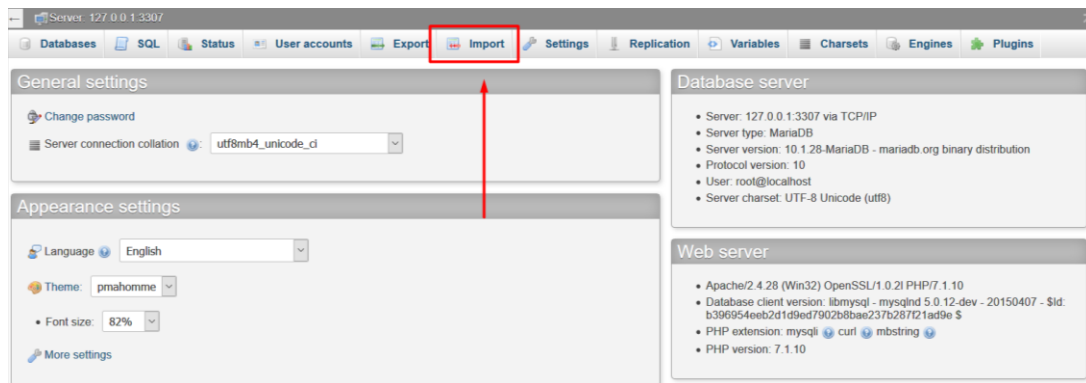


Database

Deployment

To deploy the database, you have the script located in this same directory, file named library.sql.

To deploy, it's important to install first MySQL to the server. You can download here www.mysql.com/downloads/. We recommend to use phpMyAdmin to access to MySQL, download here www.phpmyadmin.net/downloads/.



once the software is installed, access to phpMyAdmin and import the script.

Importing into the current server

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in `[format].[compression]`. Example: `.sql.zip`

Browse your computer: No s'ha seleccionat cap fitxer. (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file:

Partial import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be a good way to import large files, however it can break transactions.)

Skip this number of queries (for SQL) starting from the first one:

Other options:

☒ Enable foreign key checks

Important to mark the checkbox Enable foreign key checks in other options section before the import.

After that, in the source code you have a file named connection_data.inc (**NO** connection_data2.inc), here change the data form yours. Here you will need the place were is located, the name of the database (We recommend "library") and the password to access.

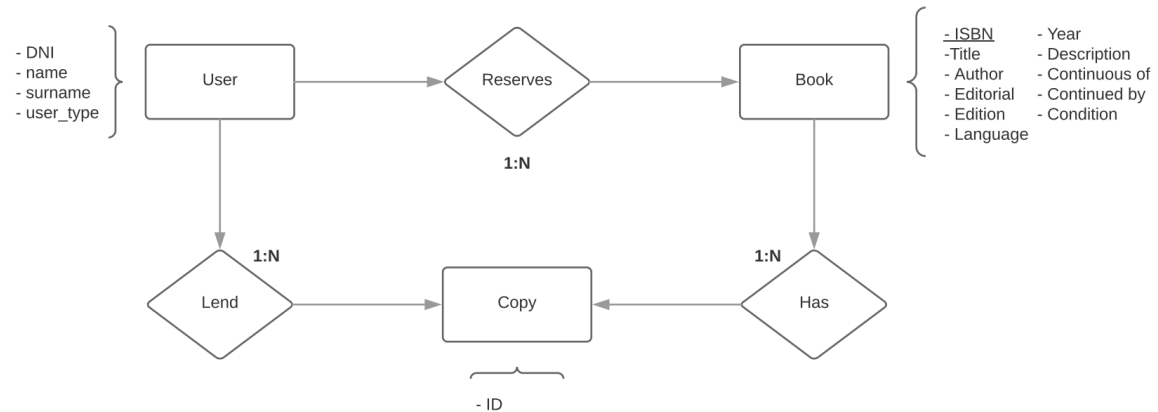
```
2 <?php
3 //connection data
4 $mysql_server = "localhost:3307";
5 $mysql_login = "root";
6 $mysql_pass = "AB.Root18";
7 ?>
```

After this steps, the database and The Library may work correctly.



Diagrams and design

The basic idea of the ER design of the database is:



Data dictionary

Here you have the data dictionary of the database.

Book table

Column	Type	Null	Default
isbn (<i>Primary</i>)	varchar(13)	No	
title	varchar(80)	No	
author	varchar(60)	No	
editorial	varchar(50)	No	
edition	varchar(2)	Yes	<i>NULL</i>
year	varchar(4)	No	
category	varchar(20)	Yes	<i>NULL</i>
language	varchar(30)	No	
description	varchar(5)	Yes	<i>NULL</i>
book_condition	varchar(7)	Yes	<i>NULL</i>
continuous_of	varchar(13)	Yes	<i>NULL</i>
continued_by	varchar(13)	Yes	<i>NULL</i>

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	isbn	24	A	No

Copy table

Column	Type	Null	Links to
id_copy (<i>Primary</i>)	int(11)	No	
isbn	varchar(13)	No	book -> isbn

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	id_copy	46	A	No
FK_isbn_copy	BTREE	No	No	isbn	46	A	No



Lend table

Column	Type	Null	Links to
id_lend (<i>Primary</i>)	int(11)	No	
start_time_lend	date	No	
dni	varchar(10)	No	users -> dni
id_copy	int(11)	No	copy -> id_copy
returned	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	id_lend	3	A	No
FK_copy_lend	BTREE	No	No	id_copy	3	A	No
FK_dni_lend	BTREE	No	No	dni	3	A	No

Reserve Table

Column	Type	Null	Links to
id_reserve (<i>Primary</i>)	int(11)	No	
start_time_reserve	date	No	
dni	varchar(10)	No	users -> dni
id_copy	int(11)	No	copy -> id_copy

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	id_reserve	2	A	No
FK_dni_reserve	BTREE	No	No	dni	2	A	No
FK_copy_reserve	BTREE	No	No	id_copy	2	A	No

Users Table

Column	Type	Null	Default
dni (<i>Primary</i>)	varchar(10)	No	
name	varchar(20)	No	
surname	varchar(30)	No	
pass	varchar(8)	No	
user_type	int(1)	No	
phone_number	varchar(9)	No	
city	varchar(20)	Yes	NULL
postal_code	varchar(11)	Yes	NULL
email	varchar(30)	No	
direction	varchar(50)	Yes	NULL










Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	dni	4	A	No














Classes diagram











Available

Available	
	DAYSOFLEND
	LIMITBOOKS
	copyDayReserved(id_copy)
	copyAvailable(id_copy, date)
	copyLend(id_copy, date)
	copiesOfBook(isbn)
	bookAvailable(isbn, date)
	numberOfCopies(isbn)
	bookReserved(dni)

Copy

Copy	
	id_copy
	isbn
	__construct()
	insertCopyToBD()
	setIdCopy(id_copy)
	addCopy(isbn)
	getIdCopy()
	__construct2(id_copy, isbn)
	__construct1(id_copy)
	getIsbn()
	setIsbn(isbn)

MakeForm

MakeForm	
	note
	submit
	actionValue
	fields
	Nfields
	Nnote
	__construct(actionValue, submit
	addField(id, label, type, required, pattern, title, value, disabled, placeholder)
	displayForm()
	addNote(note)



MakeTable

MakeTable	
f	fileBrowse
f	dbName
f	fileUpdate
f	tableName
f	fileReturn
f	condition
f	fileDelete
f	condition2
f	mysqli
f	fieldCondition
f	registers
f	filePickUp
f	row
f	numFields
f	fieldList
m	__construct(dbName, tableName, fieldList, fileBrowse, fileUpdate, fileDelete, condition, fieldCondition, condition2, filePickUp, fileReturn)
m	connectDB(dbName)
m	paintTable()
m	paintFooter()
m	paintRow()
m	moreRegisters()
m	paintHeader()

Lend

Lend	
f	id_lend
f	id_copy
f	start_time_lend
f	returned
f	dni
m	__construct()
m	setStartTimeLend(start_time_lend)
m	insertLendToBD(isbn)
m	setIdCopy(id_copy)
m	getStartTimeLend()
m	getIdLend()
m	updateLendOfBD()
m	setDni(dni)
m	getDni()
m	__construct1(id_lend)
m	__construct5(id_lend, start_time_lend, dni, id_copy, returned)
m	getIdCopy()
m	setIdLend(id_lend)
m	setReturned(returned)
m	getReturned()



MakeFormSearch

C MakeSearchForm	
f 🔒	searched
f 🔒	findEngine
m 🌱	__construct(searched, findEngine)
m 🌱	displayForm()

Book

C Book	
f 🔒	editorial
f 🔒	continued_by
f 🔒	year
f 🔒	author
f 🔒	isbn
f 🔒	edition
f 🔒	description
f 🔒	language
f 🔒	book_condition
f 🔒	title
f 🔒	category
f 🔒	continuous_of
m 🌱	__construct()
m 🌱	getEditorial()
m 🌱	setTitle(title)
m 🌱	getAuthor()
m 🌱	getCategory()
m 🌱	getContinuousOf()
m 🌱	getIsbn()
m 🌱	getEdition()
m 🌱	getLanguage()
m 🌱	getYear()
m 🌱	setEditorial(editorial)
m 🌱	getTitle()
m 🌱	updateBookOfBD(copies)
m 🌱	setLanguage(language)
m 🌱	setContinuousOf(continuous_of)
m 🌱	setCategory(category)
m 🌱	setAuthor(author)
m 🌱	setIsbn(isbn)
m 🌱	setEdition(edition)
m 🌱	getBookCondition()
m 🌱	setBookCondition(book_condition)
m 🌱	__construct1(isbn)
m 🌱	insertBookToBD(copies)
m 🌱	getDescription()
m 🌱	setDescription(description)
m 🌱	__construct12(isbn, title, author, editorial, edition, year, category, language, description, book_condition, continuous_of, continued_by)
m 🌱	getContinuedBy()
m 🌱	setYear(year)
m 🌱	setContinuedBy(continued_by)



Reserve

C Reserve	
f	id_copy
f	id_reserve
f	start_time_reserve
f	dni
m	__construct()
m	setIdReserve(id_reserve)
m	getDni()
m	updateReserveOfBD()
m	__construct4(id_reserve, start_time_reserve, dni, id_cop
m	setIdCopy(id_copy)
m	__construct1(id_reserve
m	getIdCopy()
m	insertReserveToBD(isbn)
m	getStartTimeReserve()
m	setStartTimeReserve(start_time_reserve
m	setDni(dni)
m	getIdReserve()

ShowData

C ShowData	
f	buttonDeleteOn
f	actionDelete
f	buttonModifyOn
f	actionModify
f	NLines
f	lines
m	__construct()
m	addLine(name, data)
m	displayData()
m	addButtons(actionDelete, actionModify)



User

User	
f	user_type
f	city
f	pass
f	surname
f	name
f	phone_number
f	postal_code
f	dni
f	email
f	direction
m	__construct()
m	setDirection(direction)
m	getDirection()
m	getPostalCode()
m	getPhoneNumber()
m	setCity(city)
m	setUserType(user_type)
m	getCity()
m	getName()
m	updateUser(\$id)
m	setDni(dni)
m	getSurname()
m	getDni()
m	setSurname(surname)
m	getPass()
m	setName(name)
m	insertUserToDB(register)
m	setPostalCode(postal_code)
m	construct*(dni)
m	setEmail(email)
m	getUserType()
m	setPass(pass)
m	setPhoneNumber(phone_number)
m	__construct*(dni, name, surname, pass, user_type, phone_number, city, postal_code, email, direction)
m	getEmail()



Source code

Structure

The code of The Library is open in Creative Commons 4.0. You can download all the source code here github.com/suneobbt/library.

The name system I tried to do it the most intuitive possible, if some method doesn't have a clear name, check the PHPDocs.

As you see it in the diagram classes, we have different classes:

- **Available:** Static methods to determine if a book is available or not.
- **Book:** Store book information in objects.
- **Copy:** Store copy information in objects.
- **MakeForm:** Generate code for any edit form we need in The Library.
- **MakeSearchForm:** Generate code for any search engine form we need in The Library.
- **MakeTable:** Generate code for show data in a table.
- **Lend:** Store lend information in objects.
- **Reserve:** Store reserve information in objects.
- **ShowData:** Generate code for show the data for an item.
- **User:** Store user information in objects.

The principal pages load at the client are:

- **index.php:** Homepage.
- **pageUser.php:** Home panel for users. Depend of the permits, shows different tools and information.
- **pageListed.php:** Shows any tables with data. Generated by MakeTable.php
- **pageForm.php:** Shows forms to edit items. Generated by MakeForm.php and MakeSearchForm.php.
- **pageBrowse.php:** Shows all the information of one item. Generated by ShowData.php

I wanted to add to the manual the PHPDocs, but I didn't know how to export it. It will be here for the next version. By the way, you have all the comments and information in the source code, go and check it to my GitHub if you need it. Sorry for that.

Known bugs

- Bug 12: A user can delete him self and still working inside the website.
- Bug 15: Book condition is an attribute from copy not for book.
- Bug 23: Can not decrement number of copies of one book.
- Bug 31: when modify book form is showed, number of copies isn't displayed.



Future improvements

For version 2.0 I like to implement minimum:

- Solve the known bugs.
- Add ISBN and title to table reserves and lends.
- Sticky forms for search engines.
- Add cover images and user images (Bootstrap media content).
- Encrypt the values send by get.
- Encrypt the passwords.

Useful information

Patterns.php:

Here you will find all the patterns for html forms. Is convenient to look if you have any problem with any of the forms. Sometimes You want to put any forbidden character.

Messages (alerts - alerts.php):

Sometimes, wen you do somethings, you have an alert (wrong password or user, reserve unavailable...). This alerts have codes who are sendes by get and you can see it some times, here you have the codes for any problem you can have.

- 801: Unauthorized access.
- 802: Wrong user or password on login.
- 803: Session closed.
- 804: No copy available.
- 805: Date for reserve occupied.
- 806: Register successful.
- 807: Register unsuccessful.

Comments

I would like to implement a lot of functions more, but I didn't have more time before the deadline. (I would like to sleep more too.)

For any problem, question suggestions, please contact me by email at daw2ababotllinares@iesjoanramis.org.

I hope you enjoy the code and the project as I have enjoyed developing it.

