



**Library CT**

*LMS Solution*

**CYDEO**



## ***What does this company do?***

*Provide Highly scalable library management system (**LMS**) solution*

*They **targeted** academic, special, public, and education librarians*

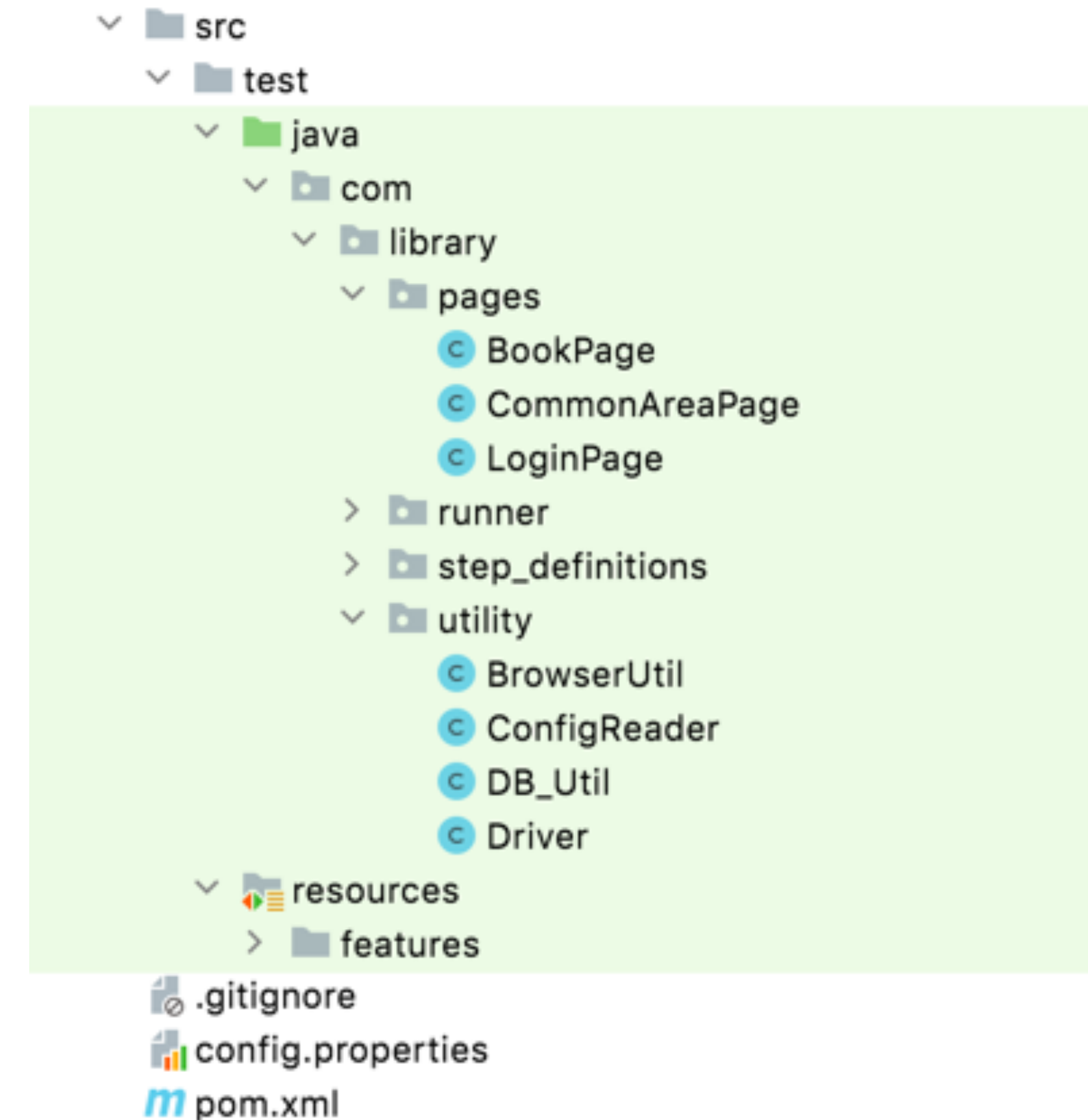
*To make the libraries **manage, circulate, and catalog** different **materials easily***

## ***What does your project's purpose? What are the features are available?***

- **User management**
  - Add / delete / update a user info [ librarian, students ]
  - Member status [Active vs Inactive]
- **Book management**
  - Add / delete / update a book
  - Book Categories: Drama, Fantasy, Humor etc
- **Search library materials**
  - Search functions
  - Filter by name, author, category, year..
- **Borrow & return books**
- **Time tracking**
  - borrowed date & time
  - Returned date & time
  - return status

# Assignment to-do list:

- **Create a new maven project – BDD Cucumber Framework**  
[GroupID: com.LibraryCT ]  
[ArtifactID: LibraryAutomation ]
- Students can work on any amount of user stories.
- Optional: If you want to use Jira, please feel free to create the user stories that you want to work in Jira.
- **Due date next Friday – 04/22/2022**



**UI credentials for Library 2:** [click here](#)

## **DB credentials for Library 2**

**Host IP:** 34.230.35.214 (SAME FOR ALL)

**Username:** library2\_client

**Password:** 6s2LQQTjBcGFfDhY

**Database name:** library2

**Port:** 3306

**JDBC url :** jdbc:mysql://34.230.35.214:3306/library2

LIBRARY SCHEMA





User Story 01

**Feature:** As a data consumer, I want the user information are stored in mySql DB correctly in users table.

**Background:**  If we can start without given we can delete background  
**Given** Establish the database connection

**Scenario:** verify users has unique IDs  
**When** Execute query to get all IDs from users  
**Then** verify all users has unique ID

**Scenario:** verify users table columns  
**When** Execute query to get all columns  
**Then** verify the below columns are listed in result

id
full_name
email
password
user_group_id
image
extra_data
status
is_admin
start_date
end_date
address

ONLY DB

users	
id	int(11) unsigned
full_name	varchar(255)
email	varchar(255)
password	varchar(255)
user_group_id	int(11)
image	text
extra_data	text
status	varchar(32)
is_admin	int(1)
start_date	date
end_date	date
address	text

## User Story 02

### Feature:

As a librarian, I want to know the amount of borrowed books

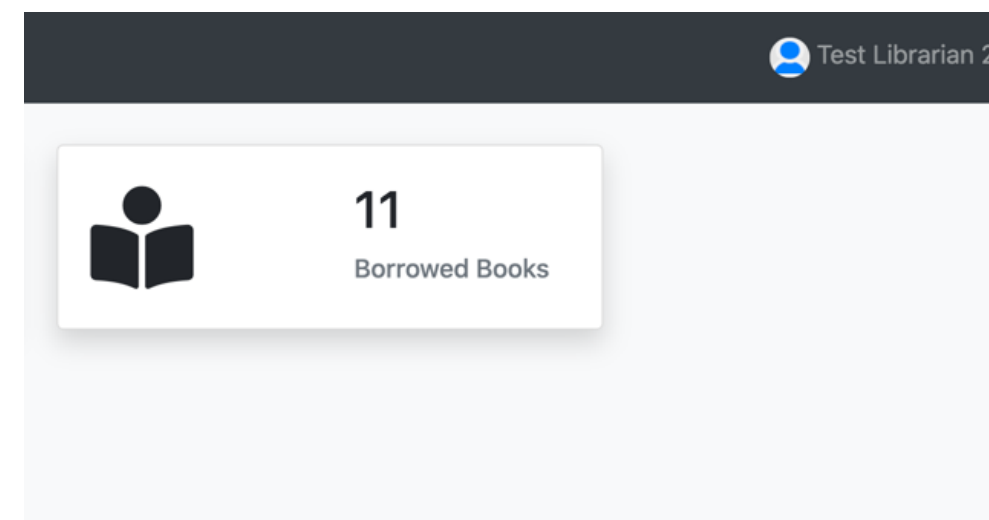
**Scenario Outline:** verify the amount of borrowed books

**Given** I am in the homepage of the library app

**When** I take borrowed books number

**Then** borrowed books number information must match with DB

### UI



### DB

1 row		
	borrowedBooks	
1		11

### Query :

```
select * from book_borrow;
```

```
select count(*) from book_borrow where is_returned=0;
```



## User Story 03

**Feature:** As a librarian, I want to know the genre of books are being borrowed the most

**Scenario:** verify the common book genre that's being borrowed

**Given** Establish the database connection  
**When** I execute a query to find the most popular book genre  
**Then** verify that "Classic" is the most popular book genre.

**Note:** *Since the most popular genre is dynamic this feature can be updated based on the data or URL.*

	name	`count(*)`
1	Action and Adventure	1889
2	Fan-Fiction	1137
3	Historical Fiction	835
4	Classic	545
5	Anthology	244
6	Drama	186
7	Fable	142
8	Horror	65
9	Comic and Graphic Novel	63
10	Fantasy	62
11	Science Fiction	56

### Query :

```
select bc.name,count(*) from book_borrow bb
      inner join books b on bb.book_id = b.id
      inner join book_categories bc on b.book_category_id=bc.id
group by name
order by 2 desc;
```

## User Story 04

**Feature:** As a librarian, I want to know who is the most popular user

**Scenario:** verify who is the most popular user who reads the most

**Given** Establish the database connection

**When** I execute a query to find the most popular user

**Then** verify "Test Student [number]" is the user who reads the most

**Query :**

```
select full_name,count(*) from users u inner join book_borrow bb on u.id = bb.user_id  
group by full_name  
order by 2 desc ;
```

User Story 05

**Feature:** As a data consumer, I want UI and DB book information are match.

**Scenario:** Verify book information with DB  
**Given** I login as a librarian  
**And** I navigate to "Books" page  
**When** I open book "Chordeiles minor"  
**Then** book information must match the Database

UI

Book Categories

ALL

Show 10 records

Search: Chordeiles minor

Actions	ISBN	Name	Author	Category	Year	Borrowed By
<a href="#">Edit Book</a>	387448631259	Chordeiles minor	Waylon Connold	Classic	2005	Test Student 22

Showing 1 to 1 of 1 entries

<<

<

1

>

>>

DB

	name	author	year
1	Chordeiles minor	Waylon Connold	2005

**Query :** `select name, author, year from books where name='Chordeiles minor';`

User Story 06

**Feature:** As a data consumer, I want UI and DB book categories match.

**Scenario:** verify book categories with DB

**Given** I log in as a librarian

**When** I navigate to the "Books" page

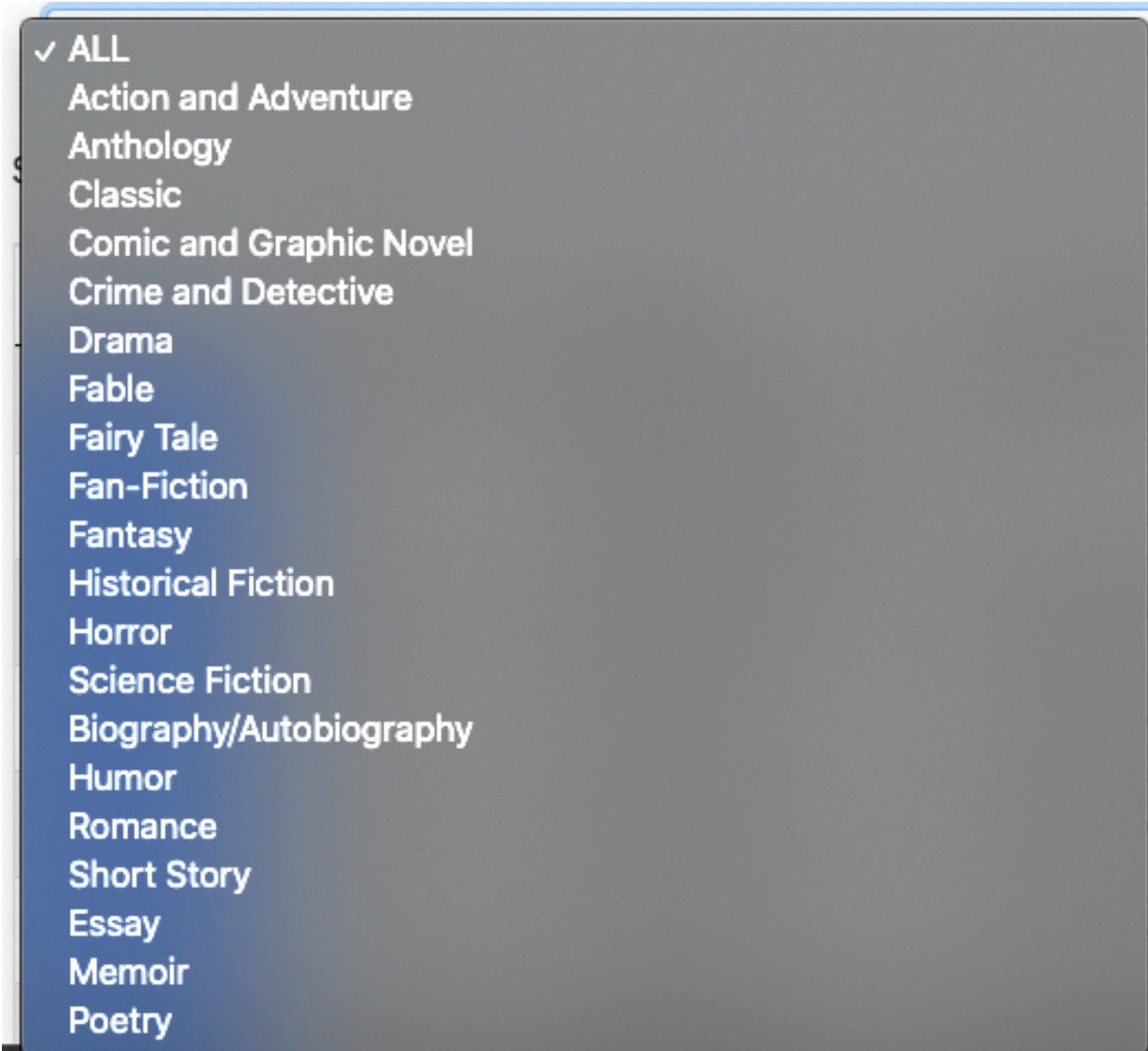
**And** I take all book categories in UI

**And** I execute a query to get book categories

**Then** verify book categories must match the book\_categories table from DB.

**Query :** `select name from book_categories;`

UI



DB

	name
1	Action and Adventure
2	Anthology
3	Classic
4	Comic and Graphic Novel
5	Crime and Detective
6	Drama
7	Fable
8	Fairy Tale
9	Fan-Fiction
10	Fantasy
11	Historical Fiction
12	Horror
13	Science Fiction
14	Biography/Autobiography
15	Humor
16	Romance
17	Short Story
18	Essay
19	Memoir
20	Poetry