In the name of GOD

third assignment report

Intrduction

this is a program to manage a simple library where a user can sign up, rent books... and librarians Supervise the actions taken in the library.
this program does the user's and librarian's requests and saves book information in itself.
it was done through objects and their methods.
but don't have a database or work via files.

Design and Implementation

design these four principle objects

- User
- Library
- Book
- librarian

now see the objects infromation

1. **User :**

attributes	methods	method explanation
username		
password	Hash password	hashs a string
	check password	Checks the correctness of the given password
	set password	sets a new password (required old password)

attributes	**methods**	**method explanation**
books list	get books	prints the list of books that were rented by the user
	return book	puts the book from book list in the library shlef
	rent book	puts the book from the library shelf in the person's book list

2. ## **Book :**

attributes	**methods**	**method job**
name	print status	prints all attributes of the book
author		
ISBN		
year of publish		
number	decreaseBook	decreases number of the book on library shelf
	increaseBook	increases number of the book on library shelf
	getNum	returns the number of book on the library shelf

3. ## **librarians**

> extended from user

attributes	**methods**	**method job**
isActive	checkActivity	returns activity status
	changeActivity	chenges activity status
	over riding checkPassword	it is also considered activity status

4. ## **library**

attributes	**methods**	**method job**
list of books	addBook	adds a book that isn't in the library with a specified amount on the library shelf

attributes	methods	method job
	removeBook	deletes a book from library
	searchBook	calls the `printStatus()` method from the book obj
	updateBook	updates year of publication of a book
	doesBookExist	checks if the book is in the library or not
	decreaseBook	deacreases the amount of book on the library self
	increaseBook	increases the amount of book on the library self
	rentBook	checks the adequate conditions and uses the `rentBook()` method of user obj
	returnBook	checks the adequate conditions and uses the `returnBook()` method of user obj
list of Users	addUser	adds a User that isn't in the library's notebook
	removeUser	deletes a user from library's notebook
	searchUser	calls the `getBook()` method from the user obj
	updateUser	updated the password by knownig the old password
	doesUserExist	checks if the user is in the library's notebook or not
list of Librarians	addLibrarian	adds a Librarian that isn't in the library's notebook
	removeLibrarian	deletes a Librarian from library's notebook
	searchLibrarian	checks the activation of the librarian
	updateLibrarian	change the activation of the librarian
	doesLibrarianExist	checks if the Librarian is in the library's notebook or not
	login	checks username and password and return the Specified value for each section

Functions that need explanation

- **User.hashPassword** :
  uses the **MD5** algorithm to hash passwords that is a secure way to save passwords in the database
  when the program wants to check the password, hashes an input password and then compares it with the hash string that was stored before.
- **library.login** :Checks the existence of user and validity of the password.

  ## return tabel :

condition	return number
user with correct password	2
user with wrong password	1
librarian with wrong password	-1
librarian with correct password	-2
else	0

and handles the value of this return in the main class and then login to user or librarian page

- Books are stored in the library in hash map because each one have specefied **ISBN** code . Search, update, add... books work based on this property of books. This is the case for users and librarians according to their **usernames**.
- This program use swing GUI to get input and show outputs. this function `JOptionpane()` creates a new window and does some stuff.
- Implements encapsulation to protect data. so initials almost every variable to private mode so can access them only with that class method. but if initials a variable public it can be changed anywhere and can't check the validity of the data or can be dangerous for important data.
- In each function, for each action, the necessary logical conditions for the inputs are checked first, and then the given command is performed

Testing and Evaluation:

- For testing, this program we set a default librarian and sign in and test all of the functions.
- The result is the program does well for each case but have a little bug so was fixed with no difficulty.
- **special case**: when a user inputs a non-numeric string when the program wants a numeric string it crashed because of the `Integer.parseInt()` function so it needed to use this way

  > `try{} catch{}`

  to detect crashes and handle them.

Conclusion:

with the help of this program, we can meet the needs of a library to manage.
but this program needs a more beautiful user interface and a strong database or a appropriate server to have a connection between libraries to can be used in practical conditions.