v1.4 Student Project LIBRARY4J

CLASS ING1-GI • 2021-2022
AUTHOR Romuald GRIGNON
E-MAIL rgn@cy-tech.fr

DESCRIPTION

Create an application to manage a stock of cultural items such as books, and manage users who borrow these books.
The main goal is to be able to create a graphic application you can use with your mouse. Your application can browse and display the books from your database, enabling filtering to narrow the results.
This project will only store the user information, the books ID, and the list of loans. All the book information will be retrieved dynamically from an external service such as a REST API (the one from the "Bibliothèque Nationale de France" – BNF, seems to be a good choice).
This project will make you work more on the graphical part and the database part, so it could fit better with data management students.

MILESTONES

Project kick-off
You will get all project information (dates, evaluation criterias, report format, project presentation, help contact, …) when the project will start.
Project design (step #1)
You will provide an UML conception document (pdf format) + containing your github project address. Your document will include a forecast planning of your developments. This planning will be updated and provided for all next milestones.
Command line version (step #2)
You will show an intermediate version of your application using the command line window. It is not compulsory to have all the data model features implemented at this time. In other hand, you shall have a precise diagram of your GUI layout.
Graphic version (step #3)
You will show another intermediate version of your application using the graphic library. It must include additional features compared to the previous milestone especially if the graphic interface is not functional yet.
Almost-Final version (step #4)
Before the final delivery, an almost final version will be shown to check and adapt the last features.
Demonstration
The day of the demonstration, you will introduce the same version you have committed on your git repository (the final delivered version) even if you have added some features or fixed some bugs in-between.

PROJECT MAIN FEATURES

- An UML conception document of your application shall be delivered, showing the different class relations, especially the ones between the view, the data model, and the application controller.
- Your whole project (images, documentations, code, ...) will be stored on a github repository for three main reasons : avoid loosing anything along your application development, being able to work as a team, and share your work progress with the teachers.
- All your project assets (code, documentation) will be written in english. This is not applicable to your project report.
- You must provide a JavaDoc documentation in html format.
- The use of external libraries such as database access could be necessary.
- Your application must not crash at anytime, whatever the reason, all exceptions must be caught and handled properly. It is better to have a stable application with less functional features, than a full feature application that crashes.
- Your application must implement user authentication : you must store credentials in a proper way in your database.
- Your application must include user roles, especially the librarian, that can search/create/update users in the database from the application, assign a book to any user during any period.
- Registered users can login in and see history of their borrowings, but cannot return or borrow any book : this is up to the librarian user to do these tasks.
- The librarian will have a specific view of all current borrowings, especially if some users are late to return their books, it must be displayed on the application.
- Your application may be accessed without login in to perform a book search : the book search will connect to an external REST API in order to get all item information. This search can be filtered at least by author, title, creation date, category, number of borrowings.

POSSIBLE RESOURCES

GitHub

- https://github.com/

Design patterns (optional)

- https://fr.wikipedia.org/wiki/Patron_de_conception
- look first at : Singleton / Strategy / Composite / Observer / Visitor

Serialization

- used to save/restore java objects
- https://fr.wikipedia.org/wiki/Sérialisation

Awt / Swing / Java FX

- look for the package documentations on the official website
- https://docs.oracle.com/en/java/javase/14/docs/api/
- https://openjfx.io/

Database connection (JDBC)

- https://docs.oracle.com/javase/tutorial/jdbc/basics/connecting.html

BNF REST API

- In order to get all book information : https://api.bnf.fr/

REST API consumption (JSON format using Maven)

- Java can execute http request and parse JSON response
- https://docs.oracle.com/javase/8/docs/api/java/net/HttpURLConnection.html
- https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple