Omid Mohaghegh Doust, omid.doust2@gmail.com, last update on 2021-07-14

Full Stack Java Developer- Developing a Backend Admin for Learner's Academy project related to course: Become a back-end expert provided by simplilearn - 2021

Git Repository: https://github.com/omid0101/OmidDoust

This project was performed in 4 sprints as following:

1st Sprint: Goal was to build backend mapping objects and examine CRUD on one of them (Class Object in this case, object name:Schoolcls.java in com.entities folder).

MyBackgroundProject-1 and Sprint One folders are related to 1st Sprint.

Note: hibernate.cfg.cml file is created for connection and reading/writing operation modes definition to DB located under src/main/java folder.

hibernateUtil.java file is created for helping such connection through Hibernate and mapping related objects (classes) to the DB located in the com.hibernate package.

DBConnection.java is used for connecting to DB under Hibernate platform through JDBC Driver and Driver Manager located in the com.connection package.

------------------------------------------------------------------------------------------------------------------------------------------------------------

2nd Sprint: Create relationships between tables and test it with CRUD. Sprint two-part one demonstrates tests on class objects and other dependant objects through relational mapping located in com.hibernate package.

Sprint two-part two demonstrates advanced test on improved programming including all objects.

MyBackgroundProject-2, Sprint two-part one and Sprint two-part two folders are related to 2nd Sprint.

There are three 4 objects, accordingly 4 entities for mapping to DB through Hibernate: Class, Teacher, Subject, Student

Relationship between entities defined as following:

Class- Subject ---> ManytoMany - ManytoMany

Class- Teacher ---> ManytoMany - ManytoMany

Class- Student ---> OnetoMany - ManytoOne

This sprint folder was tested by handwritten data in the servlets. Its workflow is similar to the 3rd sprint showing below.

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3rd Sprint: Create frond-end pages for setting up master lists for each entity, assigning different objects together described per assignment, and getting entities' reports.

All were successfully done and reports of screenshot created in the folder. Assigning subject to class, teacher to class to a subject, student to a class.

Reports were generated for each entity (more than assignment requirement). Each single entity was tested individually and in combination both in MySQL data base and in the reports generated in the front-end.

MyBackgroundProject-3 and Sprint Three folders are related to 3rd Sprint.

Note: All servlets are located in com.control package. All classes (for object mapping to DB) are located in com.entities package.

3rd Sprint Flow diagram @ Front end:

home --> Go to Master List for Set up data --> Data insertion for objects in DB (Data Base) --> return to home --> Go to Assign Page

--> Assign objects /data together --> insertion into the DB --> Show Student Report, Show Class Report, Show Subject Report, Show Teacher Report --> reporting corresponding objects from the DB

3rd Sprint Flow diagram @ Backend:

index.html @ home --> Master_List.html (Go to Master List for Set up data) -->1. ClassRegisterController Servlet @ POST method, 2.TeacherRegisterController Servlet @ POST method,

3.SubjectRegisterController Servlet @ POST method, 4.StudentRegisterController Servlet @ POST method --> Mapping data to the DB through relevant entities: Schoolcls(Class objects),

Student (student objects), Teacher (teacher objects) & Subject (subject objects)

index.html @ home --> ClassSubjectAssign Servlet (Go to Assign Page) --> SubjectAssignController @ GET (retrieving data related objects from the DB) --> Assign.jsp (transferring retrieved data

to Assign.jsp and showing in the browser)

note: a filter object was implemented for preventing redundant object shows from the DB as a technique) --> Assign function --->A

A.1) Assign Subjects to Classes (or vise versa): SubjctAssignController Servlet @ POST --> Registering the assigned objects to the DB

A.2) Assign Teachers to Classes to Subjects : TeacherAssignController Servlet @ POST --> Registering the assigned objects to the DB

A.3) Assign Students to Classes : StudentAssignController Servlet @ POST --> Registering the assigned objects to the DB

index.html @ home --> Show Student Report --> StudentShowController servlet @ GET --> retrieving data related to student objects from the DB and showing in the browser

index.html @ home --> Show Class Report --> ClassShowController servlet @ GET --> retrieving data related to class objects from the DB and showing in the browser

index.html @ home --> Show Subject Report --> StubjectShowController servlet @ GET --> retrieving data related to subject objects from the DB and showing in the browser

index.html @ home --> Show Teacher Report --> TeacherShowController servlet @ GET --> retrieving data related to teacher objects from the DB and showing in the browser

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4th Sprint: Goal in this sprint in to add login and log out and sing up options for a backend admin done successfully and the test result screenshots were provided in the folder Sprint Four. It was also tested for several admins who are able to sign up, sign in, doing operations relevant to the 3rd sprint, and log out.

Also there is message of wrong credential in case and admin enter a null login or null password or wrong login or wrong password which doesn't allow to sign in.

4th Sprint Flow diagram @ Front end:

Click Sing up link @ Home page --> @Sign up page --> @Start page --> @Log out page --> @Log in page --> @Start page ...

4th Spring Flow diagram @ Backend:

Click Sing up link @ Home page --> Sign Up.jsp --> LoginRegisterController Servlet @ POST --> start.jsp --> gets data from DB and welcome to that person name + start page operations(3rd Sprint) + log out --> logout.jsp --> message " You logged out succesfully" + login lick--> login @ Home page --> login with login and password --> login.jsp

--> login info control @ login.jsp (retrieving login info data for that admin from DB and comparing with the entered login and password info)--> 1

1.1) if correct --> start.jsp --> welcome to the admin name & all operations related to start page explained in the 3rd sprint

1.2) if not correct (error message: invalid credential! please try again!) --> index.jsp @ Home page

Sincerely,

Omid Mohaghegh Doust

Enclosed.