

Report Project 2 : WebStudentBook JSF & JPA

I. Enumerate the classes and the web files that you used

Maxime and I have used way less classes in this project in comparison to the last one... We respected the **MVC Pattern** and implemented **JSF** and **JPA** for persistence and server faces.

Three classes :

- Student

As usual, the 'Student class' was used to create new students. These students were saved to the 'studentdb' database, which is a my SQL local database on our machines.

- StudentManager

StudentManager allowed us to make the link between web files and java code. This new way of programming (JSF) is really convenient and decreases a lot the number of servlets we have to implement ! StudentManager calls StudentDBUtil functions for each of its own functions in order to add, edit, view or delete a list of students and call new web files.

- StudentDBUtil

This last class picks up students from the studentdb database, deletes/edits a student or even adds a new one.

Three web files :

- List-students

List-students.xhtml displays a list of students. This list is auto-incremented, which means that if you delete the first student in the list, the second student's id will be updated in the database as id=1 and will be displayed this way in List-students.xhtml (**screenshot number x**).

This xhtml file has 2 buttons per student (edit and delete) and one general button for adding a student.

- Add-student

Add-student.xhtml can be called, as previously said, from List-Students.xhtml. We can add a student by entering his first name, last name and email. The id is autogenerated by the database and respects the waterfall flow of ids (**screenshot number y**).

Thanks to JSF, we forced the user to always enter some data : a cell can't be empty when editing a student.

We can see below that the input is **required** and if the user enters nothing, the following message will be displayed : "Enter your ----- please".

```
<div>
  <h:inputText id="fn" value="#{TheFirstName}" required="true" requiredMessage=" Enter your first name please"/>
</div>
<div>
  <h:message for="fn" style="color:red; font-weight:bold; font-size:15px;"/><br/>
</div>
```

By the way, if you ever wondered what would happen if we clicked on the "Add a new student" button and then tried to "Cancel" to come back to List-students.xhtml, we NORMALLY wouldn't be able to do so because we specified that inputs can't be let blank. Therefore, we added an argument to the "Cancel" button, which is "immediate = true", that tells to ignore some errors or constraints that we define, such as "required = true" for instance !

```
<h:commandButton style="background-color: lightpink; padding-left:15px; padding-right:15px;"
  value="Cancel" immediate="true" action="List-students" />
```

- Edit-student

Edit-student.xhtml allows us to edit a student 😊. We can change everything (first, last names and email) except the id. Same as before, you can't erase some data and let the input blank while clicking on "Edit" (except "Cancel" as previously said).

For all those web files, we added some cool css features, such as floating bubbles and shadows.

II. Database structure

Let's quickly explain the studentdb database since we are using this database for a long time :

A student is composed of an id, a first_name, a last_name and an email. Those values can't be null (Remember ? Empty cells are not allowed in .xhtml files thanks to JSF pattern) and the id is auto-incremented.

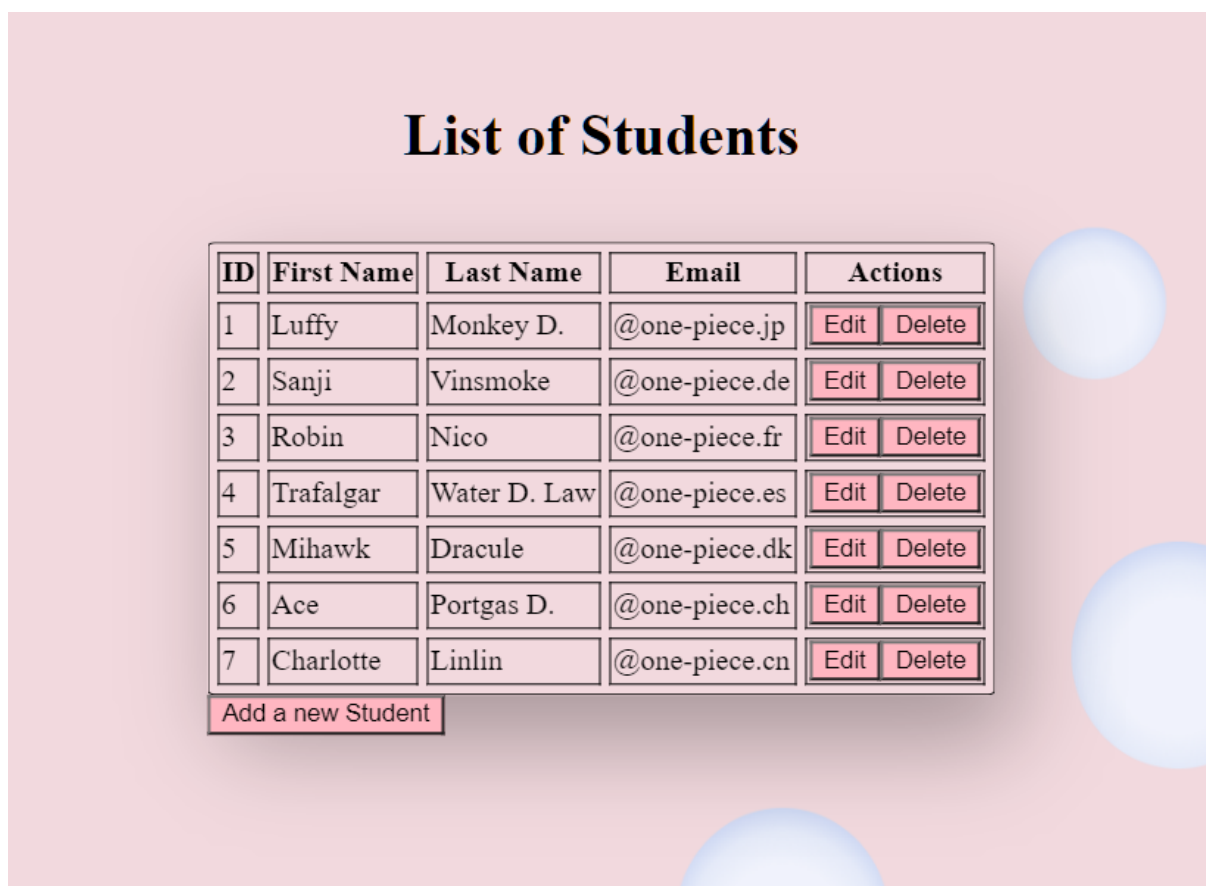
	id	first_name	last_name	email
▶	1	Luffy	Monkey D.	@one-piece.jp
	2	Sanji	Vinsmoke	@one-piece.de
	3	Robin	Nico	@one-piece.fr
	4	Trafalgar	Water D. Law	@one-piece.es
	5	Mihawk	Dracule	@one-piece.dk
	6	Ace	Portgas D.	@one-piece.ch
	7	Charlotte	Linlin	@one-piece.cn
•	NULL	NULL	NULL	NULL

III. The functionalities that you succeed to do and those that you didn't succeed.

We have implemented css, deployment, here is our report... This application allows you to add, edit and delete a student with no issue.

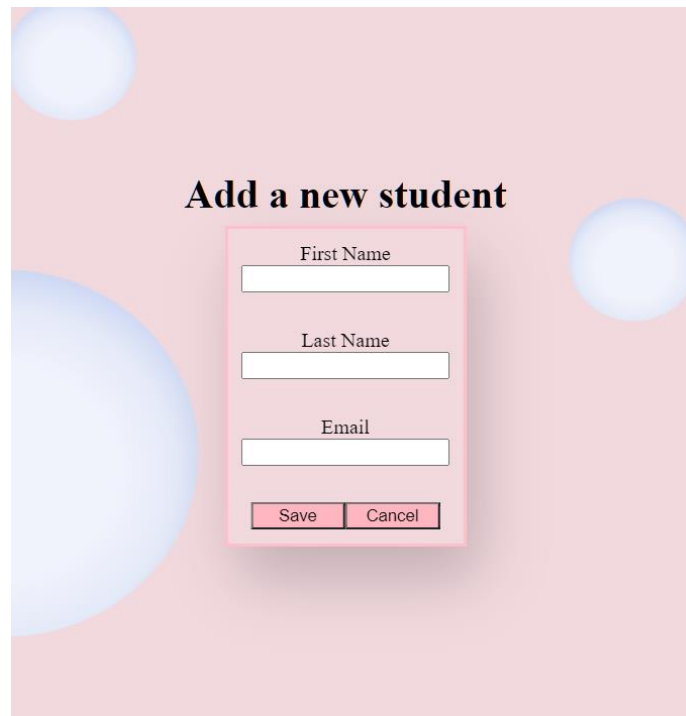
IV. Complete execution screenshots

We launch our project in deployment mode (we have been using Chrome as web browser). The list of students is displayed :



We now have several choices : we can either add a new student, edit one of the displayed students or even delete one.

Let's add a new student by clicking on the bottom on the bottom left :



Add a new student

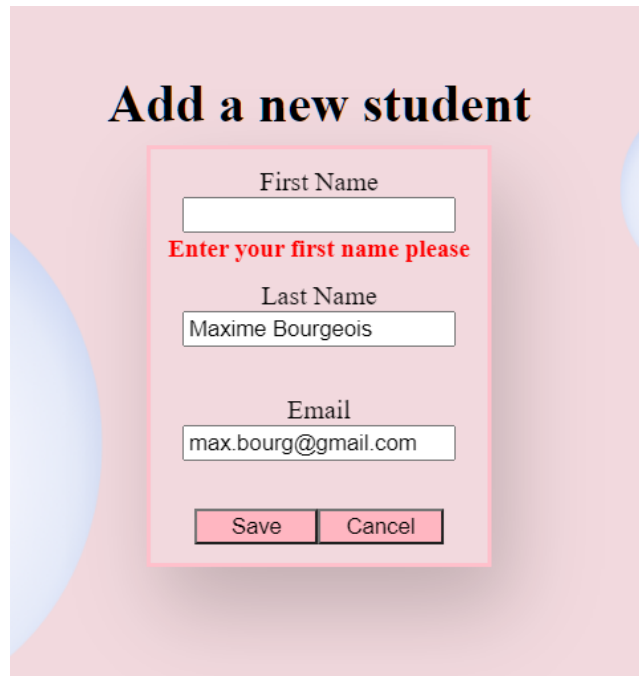
First Name

Last Name

Email

Save Cancel

Let us fill in the inputs, but we “accidentally” forget to enter the first name. We then try to submit this addition by clicking on save : we get an error.



Add a new student

First Name

Enter your first name please

Last Name

Email

Save Cancel

Let's enter a first name and submit :

Add a new student

First Name

Maxime

Enter your first name please

Last Name

Bourgeois

Email

max.bourg@gmail.com

Save

Cancel

List of Students

ID	First Name	Last Name	Email	Actions	
1	Luffy	Monkey D.	@one-piece.jp	Edit	Delete
2	Sanji	Vinsmoke	@one-piece.de	Edit	Delete
3	Robin	Nico	@one-piece.fr	Edit	Delete
4	Trafalgar	Water D. Law	@one-piece.es	Edit	Delete
5	Mihawk	Dracule	@one-piece.dk	Edit	Delete
6	Ace	Portgas D.	@one-piece.ch	Edit	Delete
7	Charlotte	Linlin	@one-piece.cn	Edit	Delete
8	Maxime	Bourgeois	max.bourg@gmail.com	Edit	Delete

Add a new Student

The new student 'Maxime Bourgeois' has indeed been added to the list of students and to the database.

Let's edit a student :

Edit a student

Student ID

8

First Name

Maxime

Last Name

Bourgeois

Email

max.bourg@gmail.com

Save

Cancel

We can for instance change his email address and save :

Edit a student

Student ID	8
First Name	Maxime
Last Name	Bourgeois
Email	max.bourg@yahoo.com
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

The email address has changed from max.bourg@gmail.com to max.bourg@yahoo.com

8	Maxime	Bourgeois	max.bourg@yahoo.com	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
---	--------	-----------	---------------------	-------------------------------------	---------------------------------------

Finally, we can delete this new student :

List of Students

ID	First Name	Last Name	Email	Actions
1	Luffy	Monkey D.	@one-piece.jp	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Sanji	Vinsmoke	@one-piece.de	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Robin	Nico	@one-piece.fr	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Trafalgar	Water D. Law	@one-piece.es	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Mihawk	Dracule	@one-piece.dk	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
6	Ace	Portgas D.	@one-piece.ch	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
7	Charlotte	Linlin	@one-piece.cn	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="button" value="Add a new Student"/>				