Denamganaï Kevin
2160104105

ソフトエアシステム特論

Software System Advance Course

REPORT

I) Design :

Overview :

The whole project has been done in JAVA 7, with a little bit of HTML and CSS with regards to the view part.

I have been using the library Spring (spring.io) to handle the basic functionnalities with regards to the handling of the database and the model-view-controller design pattern.

The servlet runs with Tomcat version 7. And I created the database using MySQL Workbench.

Design pattern : Model-View-Controller :

The whole project is entirely architecture following the POO design pattern entitle Model-View-Controller thanks to the Spring MVC Framework.

Here is a Use Case Diagram that explains the inner working of the project :

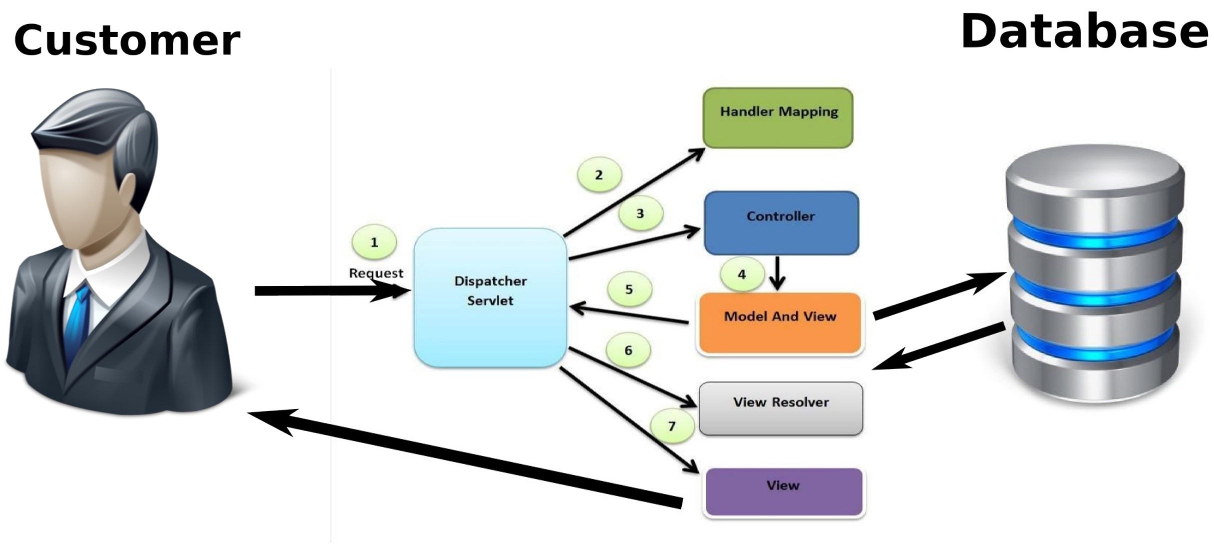


Illustration 1: Use Case Diagram

Class Diagram :

Here is the class diagram of the whole project. If you wish to see it more in depth, please refer yourself to the file ClassDiagram.pdf attached.

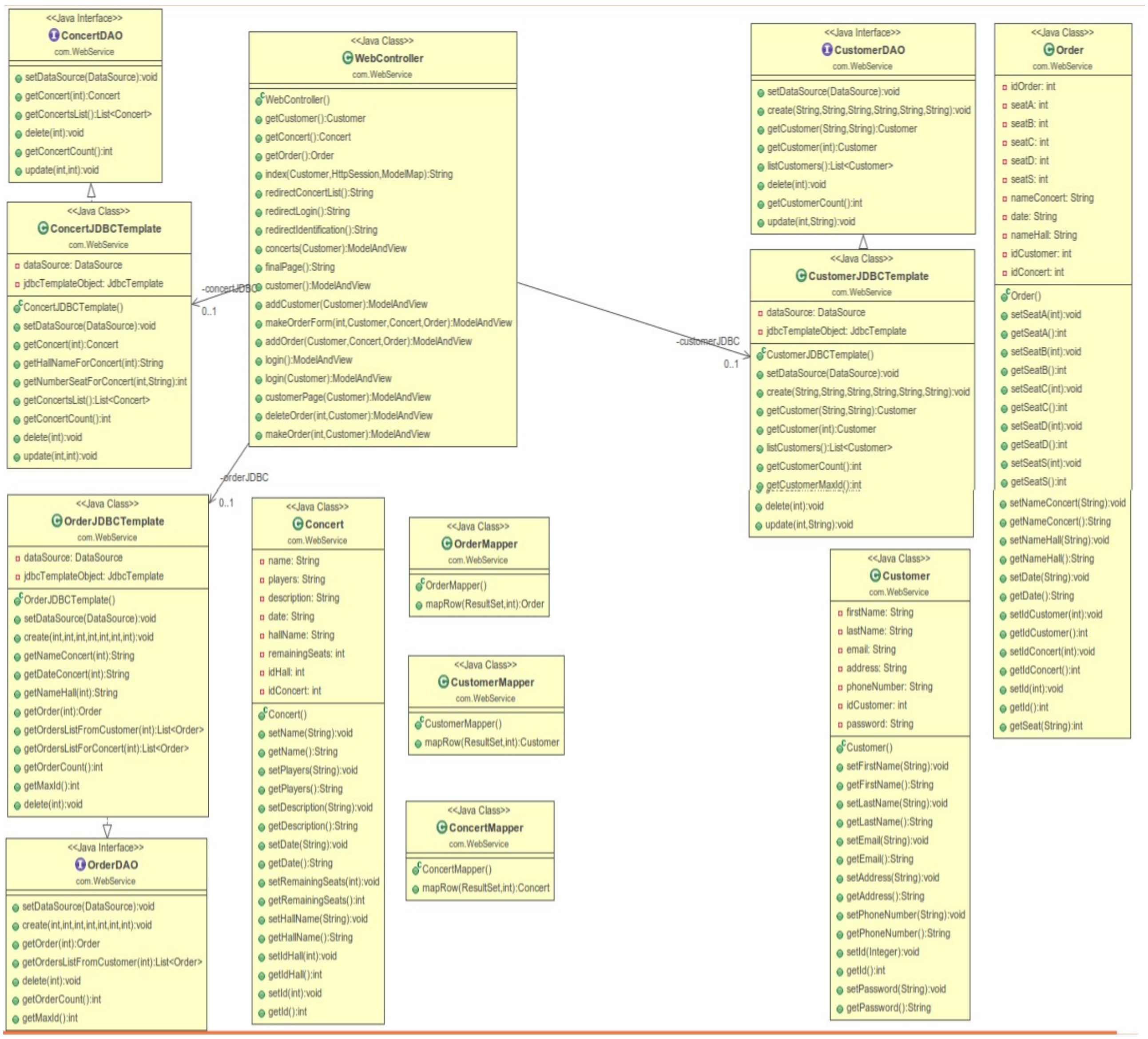


Illustration 2: Class Diagram

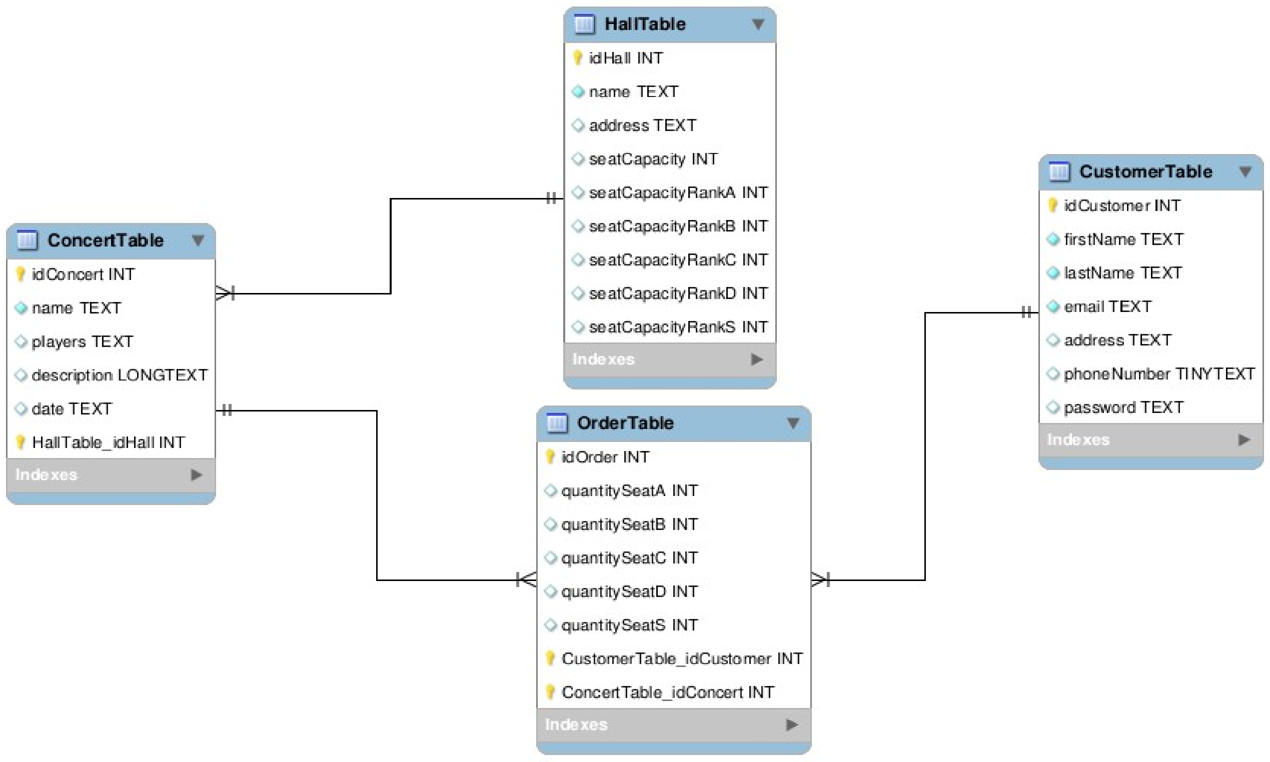
The classes Order, Concert and Customer are a direct Object-Oriented mapping of the information stored in the database, with some little extensions in order to make it more easier to work with at runtime.

The WebController class is the main port of the project that binds together the user requests, the “Model and View” aspect seen in the Use Case diagram and main model embodied by the database.

Database : EER Diagram :

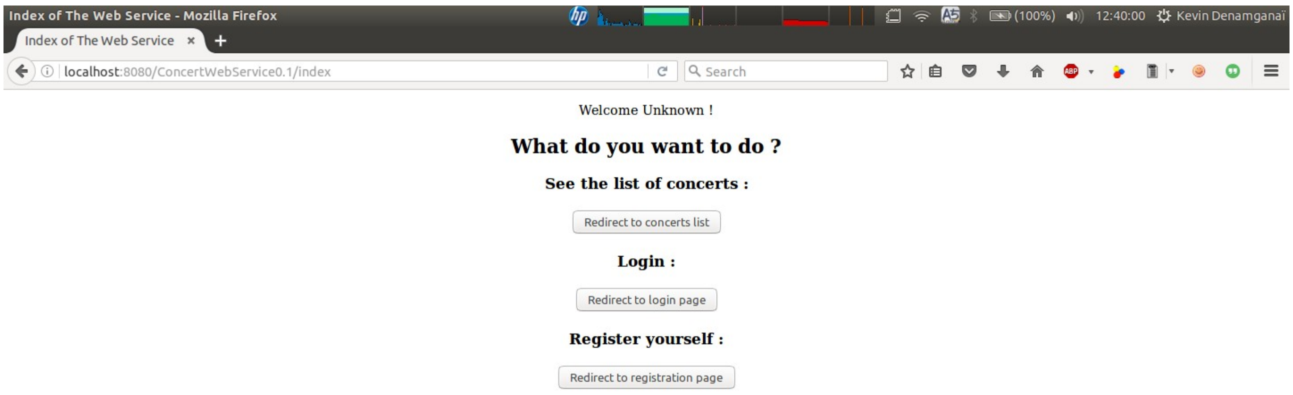
The database has been built using MySQL Workbench. Here is the EER diagram :

Illustration 3: EER Diagram



II) Sequence – Demo :

When the customer access the page for the first time, the customer is offered multiple choices. Here is the view :


Index of The Web Service - Mozilla Firefox

localhost:8080/ConcertWebService0.1/index

Welcome Unknown !

What do you want to do ?

See the list of concerts :

Redirect to concerts list

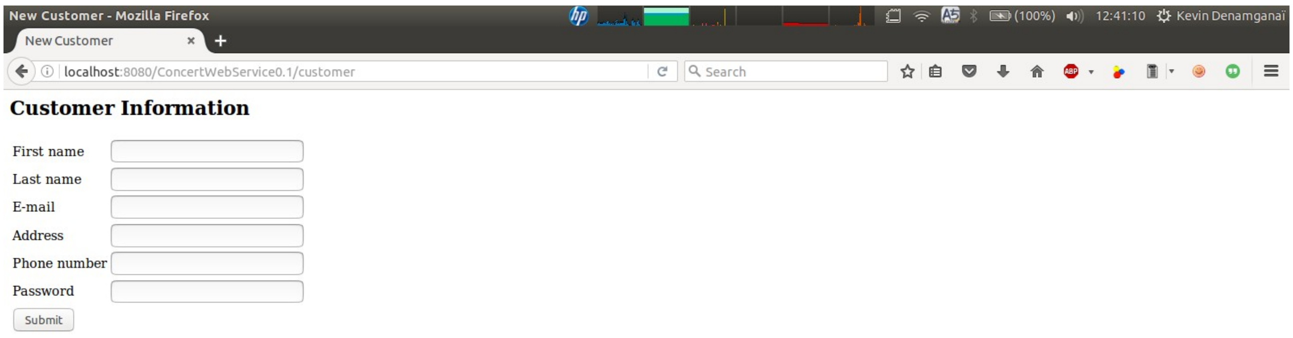
Login :

Redirect to login page

Register yourself :

Redirect to registration page

For the purpose of demonstration, let us assume that he/she decide to go on with the **registration**. Here is the nex view :


New Customer - Mozilla Firefox

localhost:8080/ConcertWebService0.1/customer

Customer Information

First name

Last name

E-mail

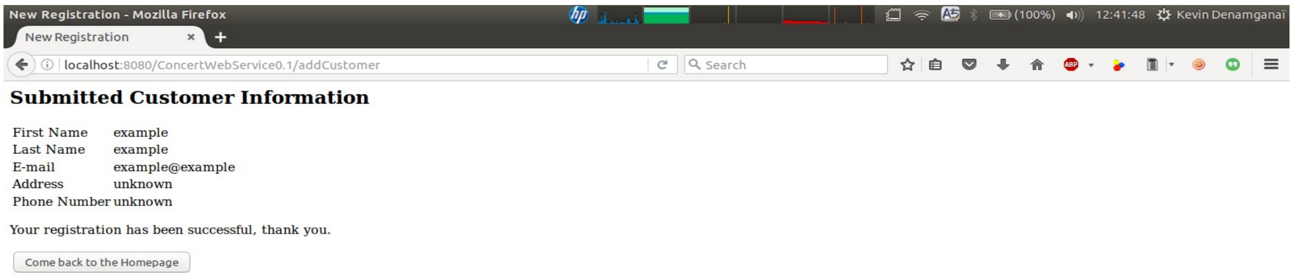
Address

Phone number

Password

Submit

If the registration is successful, here is the next view :


New Registration - Mozilla Firefox

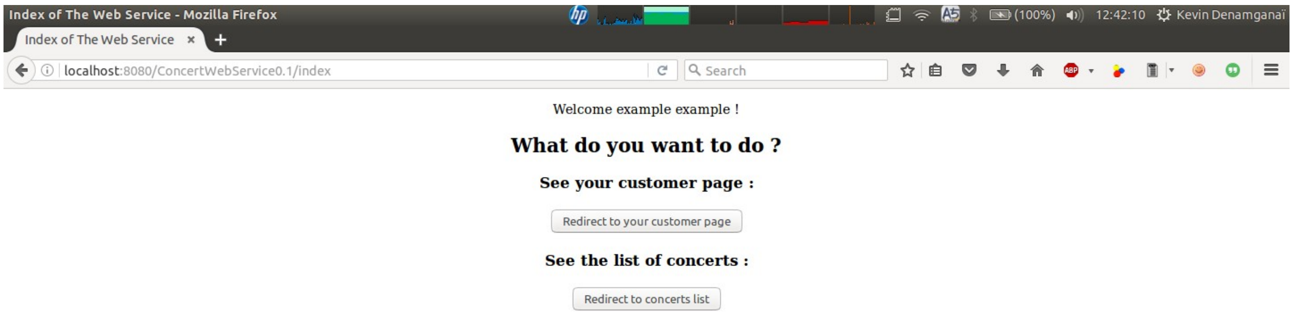
localhost:8080/ConcertWebService0.1/addCustomer

Submitted Customer Information

First Name	example
Last Name	example
E-mail	example@example
Address	unknown
Phone Number	unknown

Your registration has been successful, thank you.

There is only one choice here, it leads to that next page :


Index of The Web Service - Mozilla Firefox

localhost:8080/ConcertWebService0.1/index

Welcome example example !

What do you want to do ?

See your customer page :

See the list of concerts :

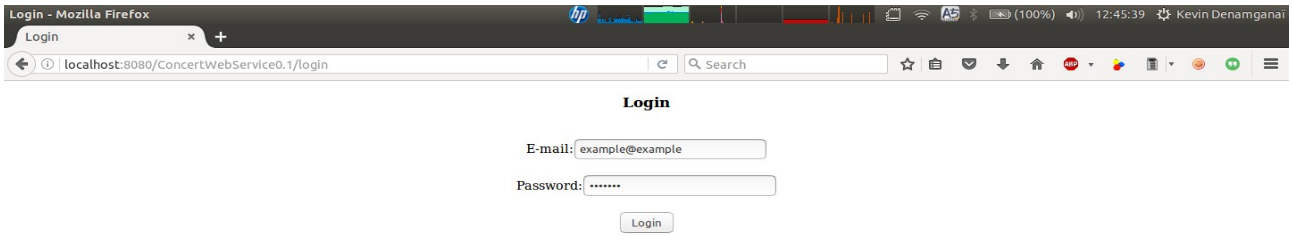
It is the same page as the beginning, but this time it has been adapted since the customer is now logged in.

By the way, the choice to resort on **login process** instead of **deletion code** has been made because it has been dimmed more trustwor, for different reasons :

1) A deletion code can be lost. Thus it makes it not very viable.

2) Asking for a deletion code from the customer means that we have to handle carefully an other input from the customer and a lot of difficulties could appear. Whereas handling a login process is way much more lightweight with regards to those constraints.

Let us come back on our demonstration. If the customer were to choose to proceed to login at the beginning, he/she would have ended up on that following view :



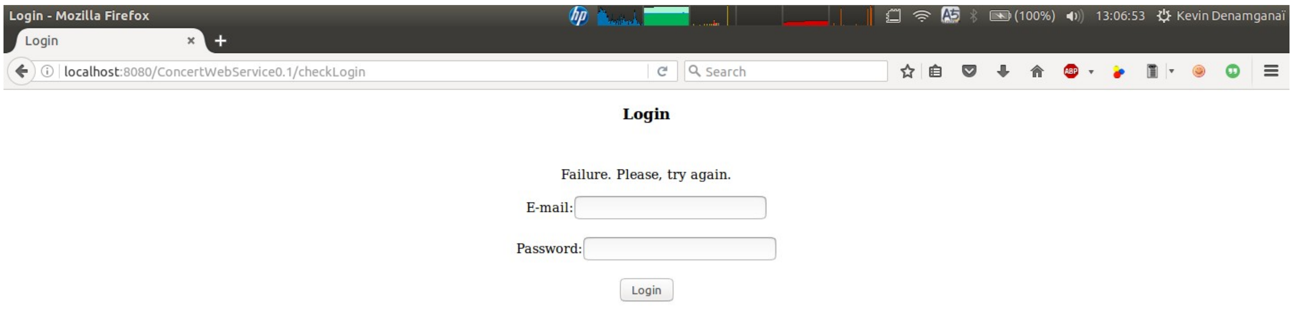
Login

E-mail: example@example

Password: •••••••

Login

If the login attempt is unsuccessful, the following view is displayed :



Login

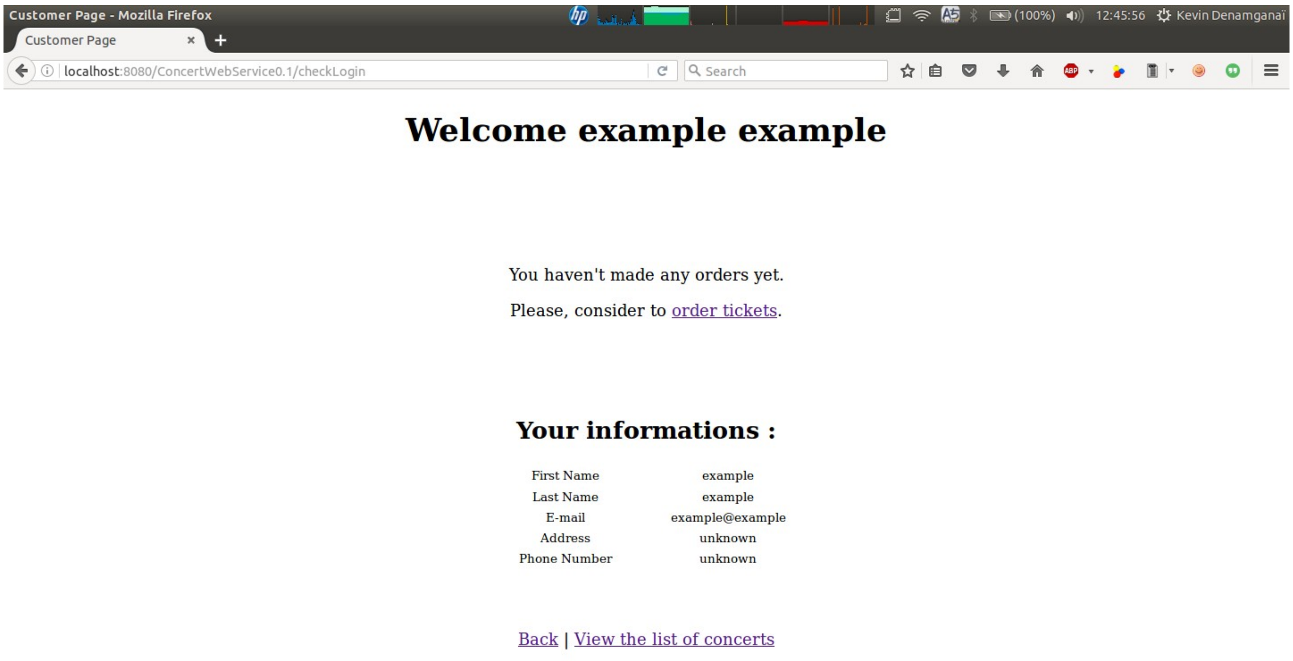
Failure. Please, try again.

E-mail:

Password:

Login

On the other case, if the login attempt is successful, the customer is redirected to his/her customer home page :


Customer Page - Mozilla Firefox
Customer Page
localhost:8080/ConcertWebService0.1/checkLogin
Search
12:45:56
Kevin Denamganaï

Welcome example example

You haven't made any orders yet.

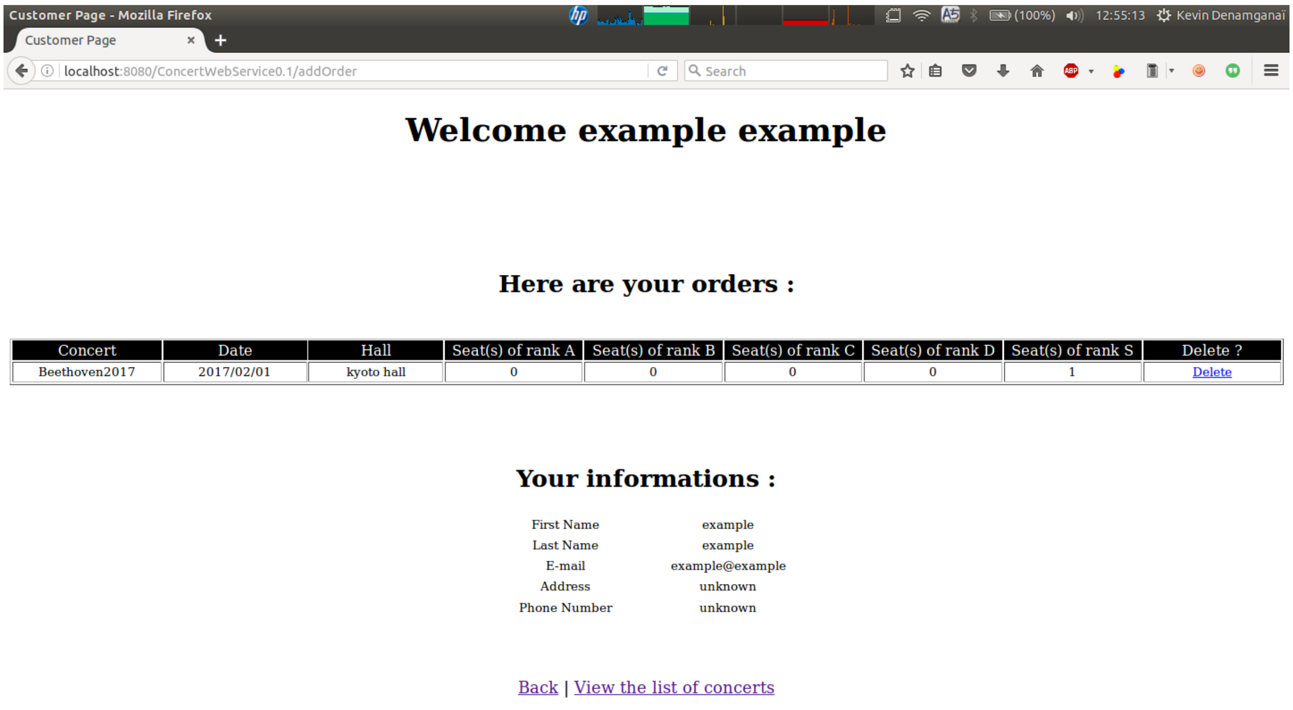
Please, consider to order tickets.

Your informations :

First Name	example
Last Name	example
E-mail	example@example
Address	unknown
Phone Number	unknown

Back | View the list of concerts

The customer can review his/her information and the list of orders that he/she has made. Here is the view displayed once the customer has made some orders :


Customer Page - Mozilla Firefox
Customer Page
localhost:8080/ConcertWebService0.1/addOrder
Search
12:55:13
Kevin Denamganaï

Welcome example example

Here are your orders :

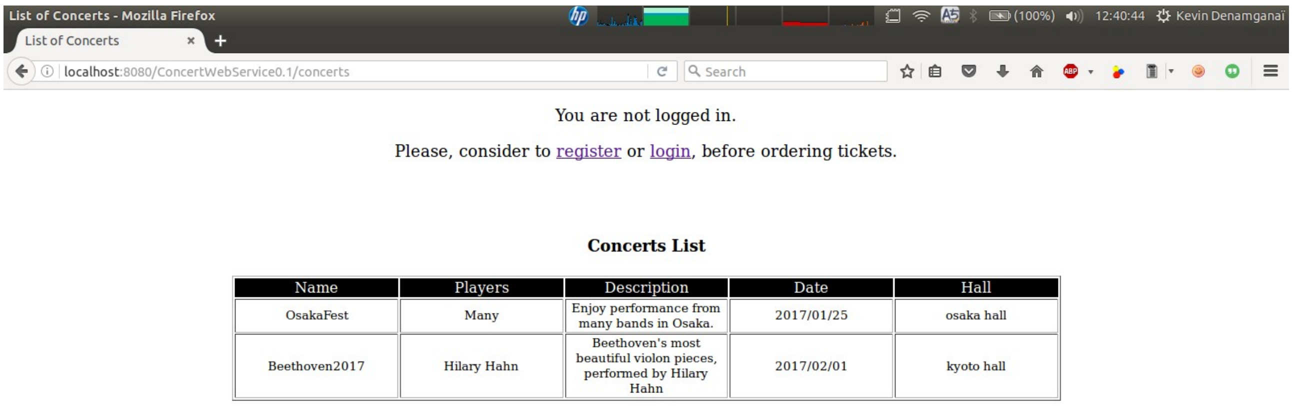
Concert	Date	Hall	Seat(s) of rank A	Seat(s) of rank B	Seat(s) of rank C	Seat(s) of rank D	Seat(s) of rank S	Delete ?
Beethoven2017	2017/02/01	kyoto hall	0	0	0	0	1	Delete

Your informations :

First Name	example
Last Name	example
E-mail	example@example
Address	unknown
Phone Number	unknown

Back | View the list of concerts

In order to make some orders, the customer must go throught the list of concerts that is displayed on the next view :


List of Concerts - Mozilla Firefox
List of Concerts
localhost:8080/ConcertWebService0.1/concerts
Search
12:40:44
Kevin Denamganaï

You are not logged in.

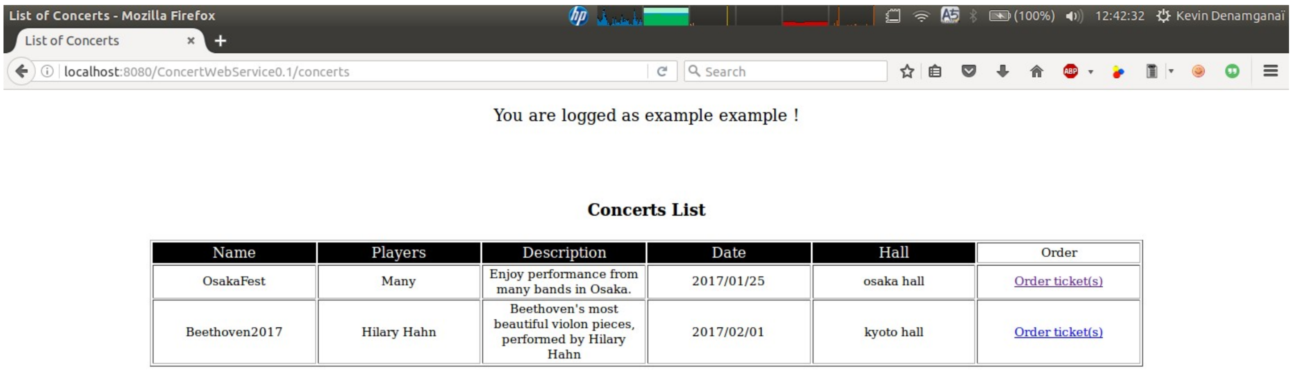
Please, consider to register or login, before ordering tickets.

Concerts List

Name	Players	Description	Date	Hall
OsakaFest	Many	Enjoy performance from many bands in Osaka.	2017/01/25	osaka hall
Beethoven2017	Hilary Hahn	Beethoven's most beautiful violon pieces, performed by Hilary Hahn	2017/02/01	kyoto hall

Back | Your Customer Page

As you can see above, this is the view related to a customer that is not logged in yet. That is to say that there are no possibility for that customer to make any order. But, once the customer is logged, the view is adapted to that following view :


List of Concerts - Mozilla Firefox
List of Concerts
localhost:8080/ConcertWebService0.1/concerts
Search
12:42:32
Kevin Denamganaï

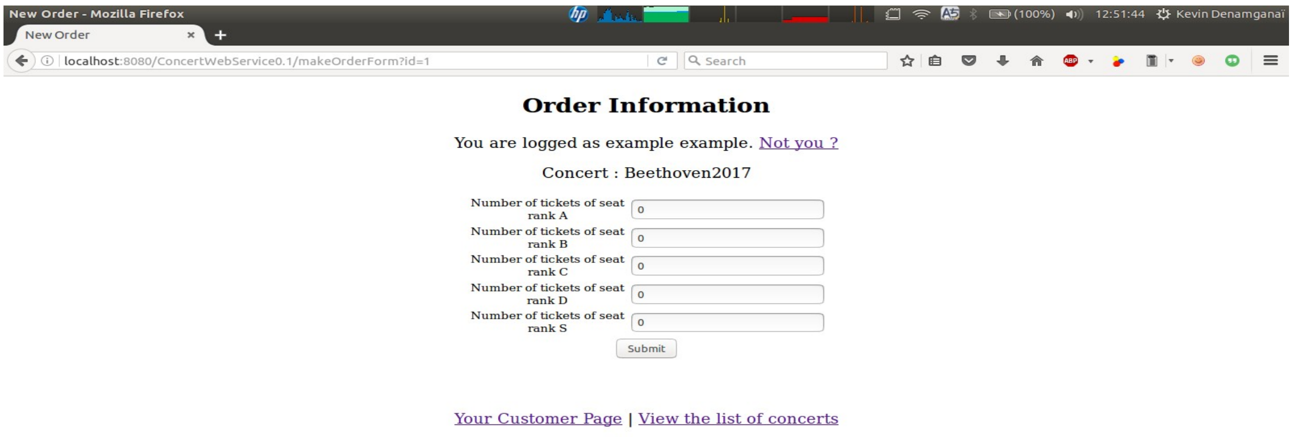
You are logged as example example !

Concerts List

Name	Players	Description	Date	Hall	Order
OsakaFest	Many	Enjoy performance from many bands in Osaka.	2017/01/25	osaka hall	Order ticket(s)
Beethoven2017	Hilary Hahn	Beethoven's most beautiful violon pieces, performed by Hilary Hahn	2017/02/01	kyoto hall	Order ticket(s)

Back | Your Customer Page

Some clickable links appear on the last column of the table of concerts. If the customer wants to make an order, he/she just has to click on one of those in order to arrive at the following view :


New Order - Mozilla Firefox

localhost:8080/ConcertWebService0.1/makeOrderForm?id=1

Order Information

You are logged as example example. Not you ?

Concert : Beethoven2017

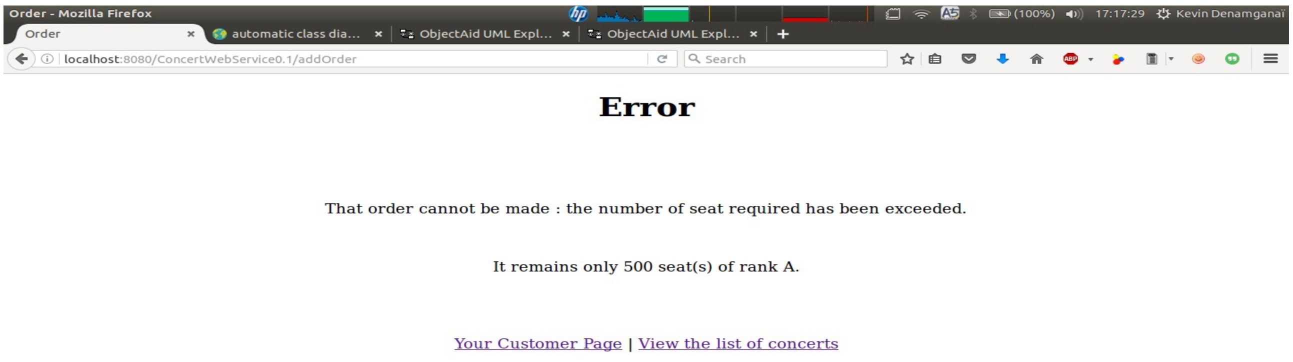
Number of tickets of seat rank A	0
Number of tickets of seat rank B	0
Number of tickets of seat rank C	0
Number of tickets of seat rank D	0
Number of tickets of seat rank S	0

Submit

Your Customer Page | View the list of concerts

Assuming that the chosen concert has not occurred yet, only in that case, the user is able to fill in an order. Otherwise, he is told that an error has occurred and that he/she cannot fill in an order for that concert.

If the number of seats requested exceeds the number of seats available, then the following view is displayed :


Order - Mozilla Firefox

localhost:8080/ConcertWebService0.1/addOrder

Error

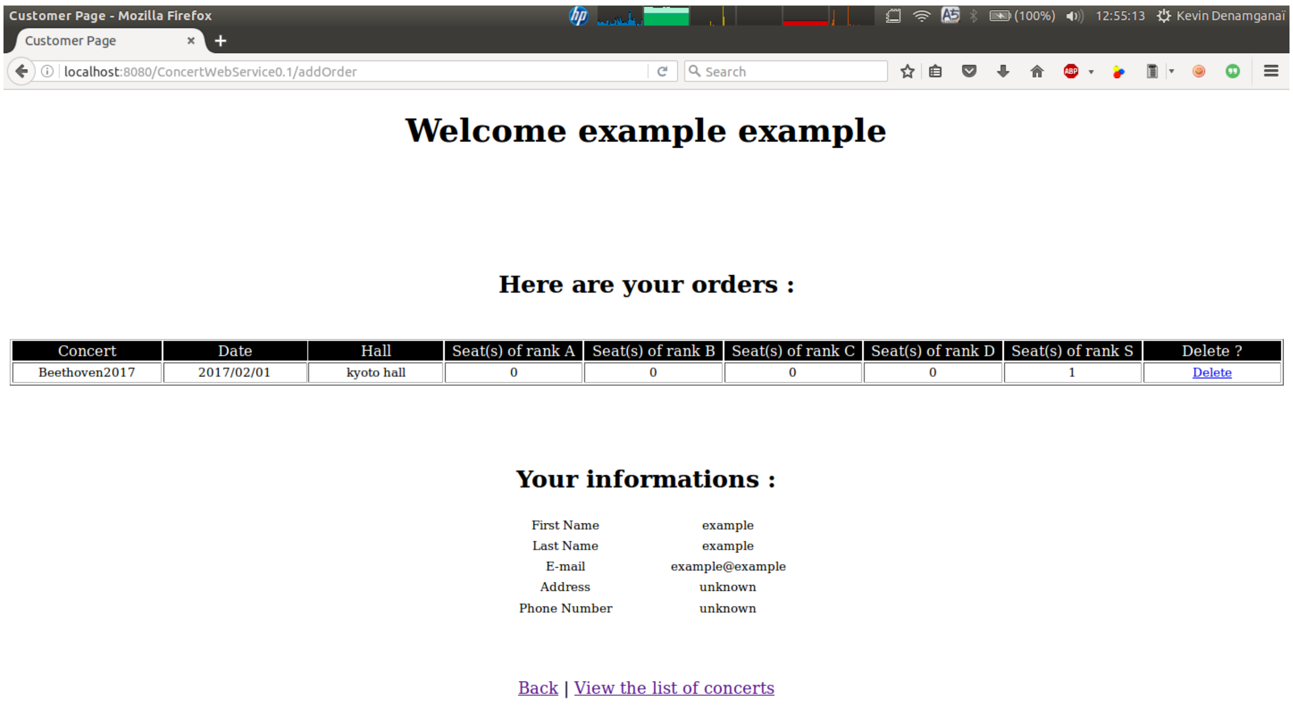
That order cannot be made : the number of seat required has been exceeded.

It remains only 500 seat(s) of rank A.

Your Customer Page | View the list of concerts

For the purpose of the demonstration, the customer has required 1000 seats of rank A for the concert OsakaFest that only provides 500 seats. When such error is produced, the customer is offered the choice to try again by going through the list of concerts once again.

If a correct number of seat is requested, for every rank, then the following view is displayed :



Customer Page - Mozilla Firefox

localhost:8080/ConcertWebService0.1/addOrder

Welcome example example

Here are your orders :

Concert	Date	Hall	Seat(s) of rank A	Seat(s) of rank B	Seat(s) of rank C	Seat(s) of rank D	Seat(s) of rank S	Delete ?
Beethoven2017	2017/02/01	kyoto hall	0	0	0	0	1	Delete

Your informations :

First Name	example
Last Name	example
E-mail	example@example
Address	unknown
Phone Number	unknown

Back | View the list of concerts

It is the customer home page from which he/she can review his/her information and the list of order made.

It can also delete some orders. That deletion is only possible if and only if the date of the concert is not passed yet.