

# Vue

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package fr.insalyon.dasi.gustatif.vue;
7
8  import fr.insalyon.dasi.gustatif.metier.modele.Client;
9  import fr.insalyon.dasi.gustatif.metier.modele.Commande;
10 import fr.insalyon.dasi.gustatif.metier.modele.LigneDeCommande;
11 import fr.insalyon.dasi.gustatif.metier.modele.LivreurDrone;
12 import fr.insalyon.dasi.gustatif.metier.modele.LivreurVelo;
13 import fr.insalyon.dasi.gustatif.metier.modele.Produit;
14 import fr.insalyon.dasi.gustatif.metier.modele.Restaurant;
15 import fr.insalyon.dasi.gustatif.metier.service.ServiceMetier;
16 import java.util.ArrayList;
17 import java.util.Date;
18 import java.util.List;
19 import java.util.Scanner;
20
21 /**
22  *
23  * @author Nico
24  */
25 public class IHMConsole {
26
27     private static final Scanner scanner = new Scanner(System.in);
28     private static final ServiceMetier serviceMetier = new
ServiceMetier();
29
30
31     public static void main(String[] args) {
32         String cmd;
33         Client client = null;
34         LivreurVelo livreur = null;
35         do {
36             System.out.println("Connexion client : c, Connexion
livreur velo : "
37                 + "\n, Gestionnaire drones : d, Inscription : i
ou Quitter : q");
38             cmd = scanner.nextLine();
39
40             if(cmd.equals("c")) {
41                 client = connexion();
42                 if(client != null) {
43                     vueRestaurants(client);
44                 }
45             } else if(cmd.equals("l")) {
46                 livreur = connexionLivreurVelo();
47                 if(livreur != null) {
48                     vueLivreurVelo(livreur);
49                 }
50             } else if(cmd.equals("i")) {

```

```
51         inscription();
52     } else if(cmd.equals("d")) {
53         List<LivreurDrone> drones =
54             serviceMetier.findAllLivresDrones();
55         if(drones != null) {
56             vueLivresDrones(drones);
57         }
58     } while(!cmd.equals("q"));
59 }
60
61
62 private static void inscription()
63 {
64     Client client = new Client();
65
66     System.out.println("Entrez votre nom : ");
67     client.setNom(scanner.nextLine());
68     System.out.println("Entrez votre prénom : ");
69     client.setPrenom(scanner.nextLine());
70     System.out.println("Entrez votre mail : ");
71     client.setMail(scanner.nextLine());
72     System.out.println("Entrez votre mot de passe : ");
73     client.setMotDePasse(scanner.nextLine());
74     System.out.println("Entrez votre adresse : ");
75     client.setAdresse(scanner.nextLine());
76     System.out.println("Entrez votre numéro de téléphone : ");
77     client.setTelephone(scanner.nextLine());
78
79     serviceMetier.createClient(client);
80 }
81
82 private static LivreurVelo connexionLivreurVelo()
83 {
84     String mail;
85     String motDePasse;
86
87     System.out.println("Entrez votre mail : ");
88     mail = scanner.nextLine();
89     System.out.println("Entrez votre mot de passe : ");
90     motDePasse = scanner.nextLine();
91
92     return
93         serviceMetier.findLivreurVeloByMailAndPassword(mail,motDePasse);
94 }
95
96 private static Client connexion()
97 {
98     String mail;
99     String motDePasse;
100
101     System.out.println("Entrez votre mail : ");
102     mail = scanner.nextLine();
103     System.out.println("Entrez votre mot de passe : ");
```

```

103         motDePasse = scanner.nextLine();
104
105         return
106     }
107
108     private static void vueRestaurants(Client client)
109     {
110         List<Restaurant> restaurants =
111             serviceMetier.findAllRestaurants();
112
113         String input, numR;
114         do {
115             System.out.println("Choisir un restaurant : r,
116                               Retour/Déconnexion : d, quitter : q");
117             input = scanner.nextLine();
118
119             if(input.equals("r")) {
120                 System.out.println("Entrez le numéro du restaurant
121                                   pour le sélectionner :");
122                 for(int i = 0; i < restaurants.size(); i++) {
123                     Restaurant r = restaurants.get(i);
124                     System.out.println("\t" + r.getId() + " - " +
125                                       r.getDenomination());
126                 }
127                 numR = scanner.nextLine();
128
129                 try {
130                     Restaurant restaurant =
131                         serviceMetier.findRestaurantById(
132                             new Long(Integer.parseInt(numR)));
133                     if(restaurant != null) {
134                         vueRestaurant(restaurant, client);
135                     }
136                 } catch(NumberFormatException e) {
137                     }
138             } else if(input.equals("q")) {
139                 System.exit(0);
140             }
141         } while(!input.equals("d"));
142
143     private static void vueRestaurant(Restaurant restaurant, Client
144     client)
145     {
146         Commande commande = new Commande(restaurant, client);
147         List<Produit> produits = restaurant.getProduits();
148         System.out.println("Restaurant : " +
149                             restaurant.getDenomination());
150         System.out.println("Description : " +
151                             restaurant.getDescription());
152         System.out.println("Adresse : " + restaurant.getAdresse());
153         List<LigneDeCommande> lignesDeCommande = new ArrayList<>();

```

```

148
149
150     String input, p, q;
151
152     do {
153         System.out.println("Récapitulatif commande : prix = " +
154             commande.getPrix()
155             + ", poids = " + commande.getPoidsEnGrammes());
156         for (LigneDeCommande ligneDeCommande :
157             commande.getLignesDeCommande()) {
158             System.out.println("\t" +
159                 ligneDeCommande.getProduit().getDenomination()
160                 + " : prix unitaire = " +
161                 ligneDeCommande.getPrixUnitaire()
162                 + "€, poids unitaire = " +
163                 ligneDeCommande.getPoidsUnitaireEnGrammes()
164                 + "g, quantité = " +
165                 ligneDeCommande.getQuantite());
166         }
167         System.out.println("Ajouter un produit : p, modifier une
168             ligne : m, "
169             + "supprimer une ligne s, valider : v,
170             annuler/retour : a, quitter q");
171
172         input = scanner.nextLine();
173
174         if(input.equals("p")) {
175             System.out.println("Entrez le numéro du produit pour
176                 le sélectionner :");
177             for(int i = 0; i < produits.size(); i++) {
178                 Produit produit = produits.get(i);
179                 System.out.println("\t" + produit.getId() + " - "
180                     + produit.getDenomination()
181                     + ", prix = " + produit.getPrix() + "€,
182                     poids = "
183                     + produit.getPoids() + "g");
184             }
185             p = scanner.nextLine();
186
187             System.out.println("Entrez la quatité :");
188             q = scanner.nextLine();
189             try {
190                 Produit prod = serviceMetier.findProduitById(new
191                     Long(Integer.parseInt(p)));
192                 if(prod != null) {
193                     int n = Integer.parseInt(q);
194                     LigneDeCommande l = new LigneDeCommande(prod,n);
195                     int index = -1;
196                     for(int i=0; i < lignesDeCommande.size(); i++) {
197                         if(lignesDeCommande.get(i).getProduit().getI
198                             d() == prod.getId()) {
199                             index= i;
200                             break;

```

```

188         }
189     }
190     if(index != -1) {
191         lignesDeCommande.get(index).setQuantite(
192             lignesDeCommande.get(index).getQuantite() + l.getQuantite());
193     } else {
194         lignesDeCommande.add(l);
195     }
196 }
197 } catch(NumberFormatException e) {
198 }
199 }
200 } else if (input.equals("m")) {
201     if(!lignesDeCommande.isEmpty()) {
202         System.out.println("Entrez le numéro de la ligne à
203         modifier:");
204         for(int i = 0; i < lignesDeCommande.size(); i++) {
205             System.out.println("\t" + i + " - "
206                 + lignesDeCommande.get(i).getProduit().getDenomination()
207                 + ", quantité = " + lignesDeCommande.get(i).getQuantite());
208         }
209         String indexLigne = scanner.nextLine();
210         System.out.println("Entrez la nouvelle quantité :");
211         String ql = scanner.nextLine();
212         try {
213             lignesDeCommande.get(Integer.parseInt(indexLigne)).setQuantite(
214                 Integer.parseInt(ql));
215         } catch(NullPointerException |
216             NumberFormatException e) {
217         }
218     }
219 } else if (input.equals("s")) {
220     if(!lignesDeCommande.isEmpty()) {
221         System.out.println("Entrez le numéro de la ligne à
222         supprimer:");
223         for(int i = 0; i < lignesDeCommande.size(); i++) {
224             System.out.println("\t" + i + " - "
225                 + lignesDeCommande.get(i).getProduit().getDenomination()
226                 + ", quantité = " + lignesDeCommande.get(i).getQuantite());
227         }
228         String indexLigne = scanner.nextLine();
229         try {

```

```
        lignesDeCommande.remove(Integer.parseInt(indexLi  
gne));  
229     } catch (NullPointerException |  
        NumberFormatException e) {  
230  
231     }  
232 }  
233 } else if (input.equals("v")) {  
234     //valider :  
235     if (commande.getLignesDeCommande().size() > 0) {  
236         commande.setDateDebut(new Date());  
237         serviceMetier.createCommande(commande);  
238     }  
239     return;  
240 } else if (input.equals("q")) {  
241     System.exit(0);  
242 }  
243 commande.setLignesDeCommande(lignesDeCommande);  
244 } while (!input.equals("a"));  
245  
246 }  
247  
248 private static void vueLivreurVelo(LivreurVelo livreur) {  
249     String cmd;  
250     String noC;  
251     do {  
252         System.out.println("Afficher détails d'une commande : c,  
Retour : r ou Quitter : q");  
253         cmd = scanner.nextLine();  
254  
255         if (cmd.equals("c")) {  
256             List<Commande> commandes =  
serviceMetier.findCommandesByLivreurId(livreur.getId());  
257             System.out.println("Entrez le numéro d'une commande  
pour le sélectionner :");  
258             for (Commande c : commandes)  
259             {  
260                 String l = "\t" + c.getId();  
261                 if (c.getDateFin() == null) {  
262                     l += " - non cloturée";  
263                 } else {  
264                     l += " - cloturée";  
265                 }  
266                 System.out.println(l);  
267             }  
268             noC = scanner.nextLine();  
269             try {  
270                 Commande commande = serviceMetier.findCommandeById(  
271                     new Long(Integer.parseInt(noC)));  
272                 if (commande != null) {  
273                     detailsCommande(commande);  
274                 }  
275             } catch (NumberFormatException e) {  
276
```

```
277     }
278     } else if(cmd.equals("q")) {
279         System.exit(0);
280     }
281     } while(!cmd.equals("r"));
282 }
283
284 private static void vueLivresDrones(List<LivreurDrone> livreurs) {
285     String cmd;
286     String noC;
287     do {
288         System.out.println("Afficher détails d'une commande : c, "
289             + "Retour : r ou Quitter : q");
290         cmd = scanner.nextLine();
291
292         if(cmd.equals("c")) {
293             List<Commande> commandes = new ArrayList<>();
294             for(LivreurDrone l : livreurs) {
295                 List<Commande> liste =
296                     serviceMetier.findCommandesByLivreurId(l.getId());
297                 if(liste != null) {
298                     commandes.addAll(liste);
299                 }
300                 System.out.println("Entrez le numéro d'une commande
301                     pour le sélectionner :");
302                 for(Commande c : commandes)
303                 {
304                     String l = "\t" + c.getId();
305                     if(c.getDateFin() == null) {
306                         l += " - non cloturée";
307                     } else {
308                         l += " - cloturée";
309                     }
310                     System.out.println(l);
311                 }
312                 noC = scanner.nextLine();
313
314                 try{
315                     Commande commande = serviceMetier.findCommandeById(
316                         new Long(Integer.parseInt(noC)));
317                     if(commande != null) {
318                         detailsCommande(commande);
319                     }
320                 } catch(NumberFormatException e) {
321                 }
322             } else if(cmd.equals("q")) {
323                 System.exit(0);
324             }
325         } while(!cmd.equals("r"));
326     }
327 }
328
```



```

329     private static void detailsCommande(Commande commande) {
330         String cmd;
331         String body = "Détails de la commande n°" + commande.getId()
332             + "\n    - Date/heure creation : " +
333                 commande.getDateDebut().toString()
334                 + "\n    - Date/heure cloture : ";
335         if(commande.getDateFin() != null) {
336             body += commande.getDateFin().toString();
337         }
338         body += "\n    - Livreur : " + commande.getLivreur().getId()
339             + "\n    - Restaurant : " +
340                 commande.getRestaurant().getDenomination()
341                 + "\n    - Client : "
342                 + "\n        " + commande.getClient().getPrenom() + " "
343                 + commande.getClient().getNom().toUpperCase()
344                 + "\n        " + commande.getClient().getAdresse()
345                 + "\n        " + commande.getClient().getTelephone()
346                 + "\n    - Produits :";
347
348         for (LigneDeCommande ligne : commande.getLignesDeCommande()) {
349             body += "\n        - " + ligne.getQuantite() + " "
350                 + ligne.getProduit().getDenomination() + " "
351                 + ligne.getQuantite() + " x " +
352                 ligne.getPrixUnitaire() + "€";
353         }
354         body += "\n    - TOTAL : " + commande.getPrix() + "€";
355         System.out.println(body);
356
357         if(commande.getDateFin() ==null)
358         {
359             System.out.println("Cloturer commande : c, Retour : r ou
360                 Quitter : q");
361             cmd = scanner.nextLine();
362
363             if(cmd.equals("c")) {
364                 commande.setDateFin(new Date());
365                 serviceMetier.validateCommande(commande);
366             } else if(cmd.equals("q")) {
367                 System.exit(0);
368             }
369         }
370     }
371 }

```