

Vue

```

1  /*
2  * To change this license header, choose License Headers in Project
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
29         ServiceMetier();
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
37                 livreur velo : "
38                 + "\n, Gestionnaire drones : d, Inscription : i
39                 ou Quitter : q");
40             cmd = scanner.nextLine();
41
42             if(cmd.equals("c")) {
43                 client = connexion();
44                 if(client != null) {
45                     vueRestaurants(client);
46                 }
47             } else if(cmd.equals("l")) {
48                 livreur = connexionLivreurVelo();
49                 if(livreur != null) {
50                     vueLivreurVelo(livreur);
51                 }
52             } 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().getId() == prod.getId()) {
198                             index = i;
199                             break;
200                         }
201                     }
202                 }
203             } catch (Exception e) {
204                 //
205             }
206         }
207     } while (input != "q");
208 }

```

```

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,
253                             Retour : r ou Quitter : q");
254         cmd = scanner.nextLine();
255
256         if (cmd.equals("c")) {
257             List<Commande> commandes =
258                 serviceMetier.findCommandesByLivreurId(livreur.getId());
259             if (commandes != null) {
260                 System.out.println("Entrez le numéro d'une
261                                     commande pour le sélectionner :");
262                 for (Commande c : commandes)
263                 {
264                     String l = "\t" + c.getId();
265                     if (c.getDateFin() == null) {
266                         l += " - non cloturée";
267                     } else {
268                         l += " - cloturée";
269                     }
270                     System.out.println(l);
271                 }
272                 noC = scanner.nextLine();
273                 try {
274                     Commande commande =
275                         serviceMetier.findCommandeById(
276                             new Long(Integer.parseInt(noC)));
277                     if (commande != null) {
278                         detailsCommande(commande);
279                     }
280                 }

```

```

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

```



```

328     }
329
330
331     private static void detailsCommande(Commande commande) {
332         String cmd;
333         String body = "Détails de la commande n°" + commande.getId()
334             + "\n    - Date/heure creation : " +
335                 commande.getDateDebut().toString()
336             + "\n    - Date/heure cloture : ";
337         if(commande.getDateFin() != null) {
338             body += commande.getDateFin().toString();
339         }
340         body += "\n    - Livreur : " + commande.getLivreur().getId()
341             + "\n    - Restaurant : " +
342                 commande.getRestaurant().getDenomination()
343             + "\n    - Client : "
344             + "\n        " + commande.getClient().getPrenom() + " "
345             + commande.getClient().getNom().toUpperCase()
346             + "\n        " + commande.getClient().getAdresse()
347             + "\n        " + commande.getClient().getTelephone()
348             + "\n    - Produits :";
349
350         for (LigneDeCommande ligne : commande.getLignesDeCommande()) {
351             body += "\n        - " + ligne.getQuantite() + " "
352                 + ligne.getProduit().getDenomination() + " "
353                 + ligne.getQuantite() + " x " +
354                 ligne.getPrixUnitaire() + "€";
355         }
356         body += "\n    - TOTAL : " + commande.getPrix() + "€";
357         System.out.println(body);
358
359         if(commande.getDateFin() ==null)
360         {
361             System.out.println("Cloturer commande : c, Retour : r ou
362                 Quitter : q");
363             cmd = scanner.nextLine();
364
365             if(cmd.equals("c")) {
366                 commande.setDateFin(new Date());
367                 serviceMetier.validateCommande(commande);
368             } else if(cmd.equals("q")) {
369                 System.exit(0);
370             }
371         }
372     }
373 }

```