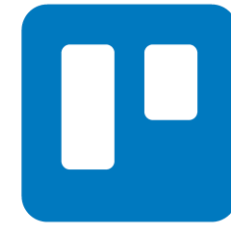


A stylized sun graphic on the left side of the slide. It features a solid yellow circle at the bottom left, with several short, curved yellow lines arching upwards and to the right from its top edge. The background is split: orange on the left and white on the right, separated by a curved line that follows the sun's path.

Présentation du projet gestion Orange en Java

Outils utilisés :



Trello



eclipse



Démarche prise :

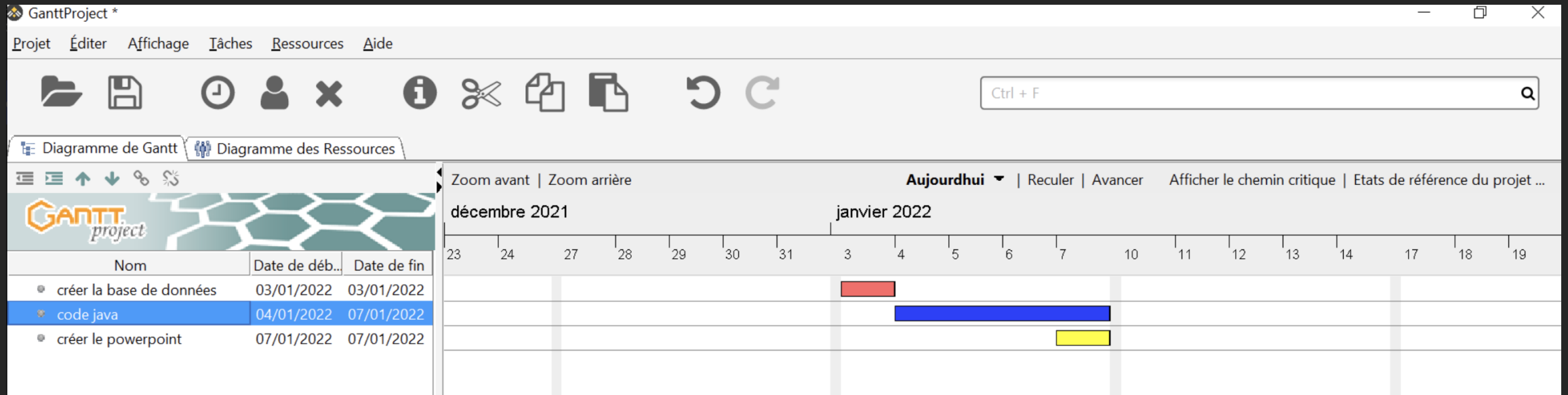
- 1) Utiliser le Discord permettant de mettre en commun nos avancés facilement
- 2) Se mettre d'accord sur quel projet nous allons nous pencher
- 3) Se répartir les tâches

Péripéties :

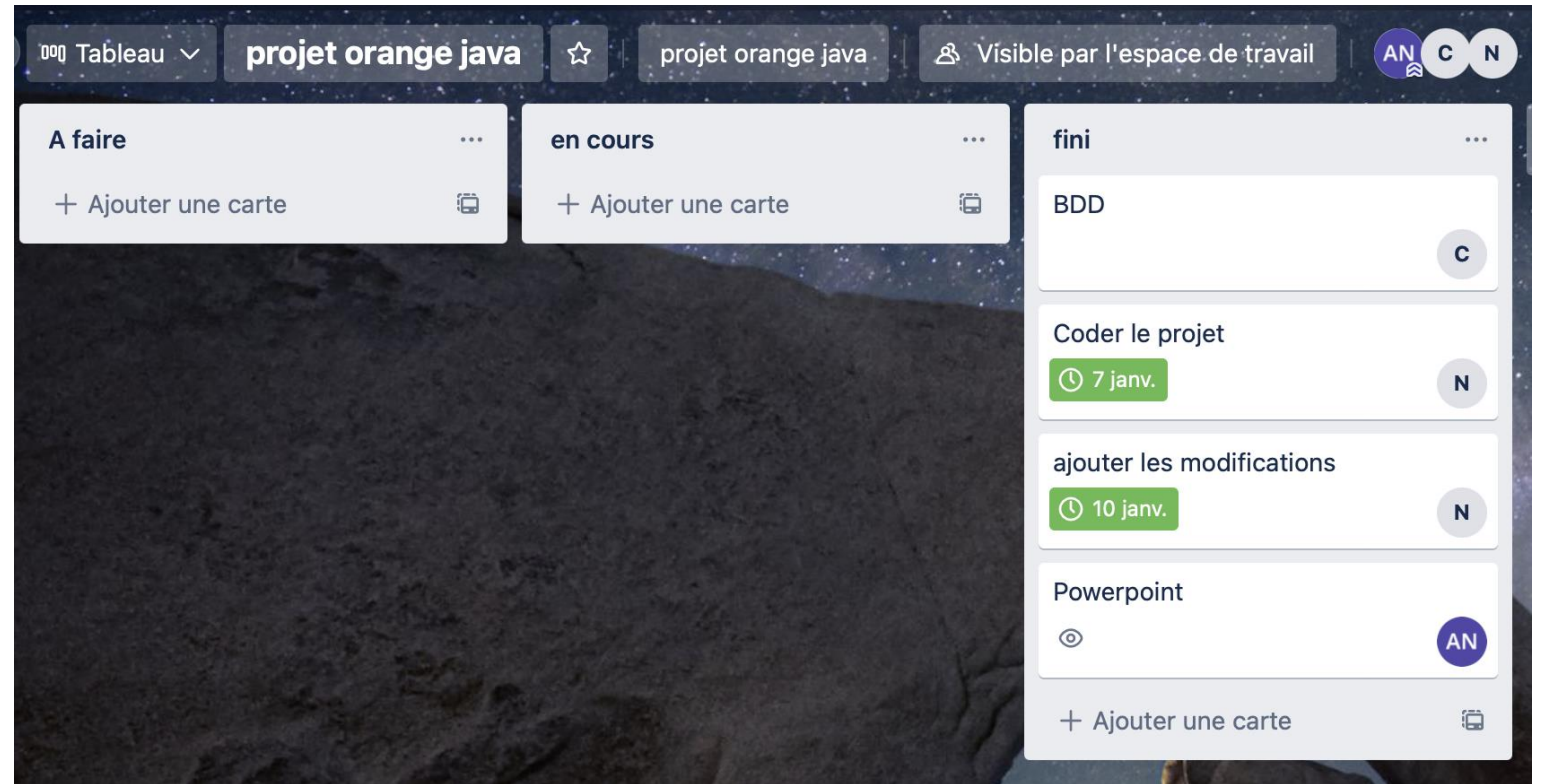
Se mettre d'accord sur
les démarches à suivre

S'organiser pour rendre
le travail dans les délais

Diagramme de Gantt :



Trello :



La base de données :

```
orange.sql
1 drop database if exists orange;
2 create database orange;
3 use orange;
4
5 create table technicien (
6     idtechnicien int(3) not null auto_increment,
7     nom varchar(50) not null,
8     prenom varchar(50) not null,
9     qualification varchar(100) ,
10    primary key (idtechnicien)
11 );
12
13 create table client (
14     idclient int(3) not null auto_increment,
15     nom varchar(50) not null,
16     prenom varchar(50) not null,
17     email varchar(50) not null,
18     tel varchar(50) not null,
19     primary key (idclient)
20 );
21
22 create table intervention (
23     idintervention int(3) not null auto_increment,
24     libelle varchar(50) not null,
25     dateintervention date not null,
26     idclient int(3) not null,
27     idtechnicien int(3) not null,
28     primary key (idintervention),
29     foreign key(idclient) references client(idclient),
30     foreign key(idtechnicien) references technicien(idtechnicien)
31 );
32
33 Create or replace view cti as (
34     select c.nom as nomcli, c.prenom as prenomcli, t.nom as nomtech, t.prenom as prenomtech, i.libelle as libelle, i.dateintervention as dateintervention
35     from technicien t, client c, intervention i
36     where c.idclient = i.idclient
37     and t.idtechnicien = i.idintervention
38 );
```

```

257 + unIntervention.getIdLibelle() + ", '"
258 + unIntervention.getDate() + "'", '"
259 + unIntervention.getIdClient() + "'", '"
260 + unIntervention.getIdTechnicien() + "'"");";
261 }
262 uneBdd.seConnecter();
263 Statement unStat = uneBdd.getMaConnexion().createStatement();
264 unStat.execute(requete);
265 unStat.close();
266 uneBdd.seDeconnecter();
267 } catch (SQLException exp) {
268     System.out.println("Erreur de requête : " + requete);
269 }
270
271 // SELECTION DE TOUTE LES INTERVENTIONS
272 public static ArrayList<Intervention> selectAllInterventions () {
273     ArrayList<Intervention> lesInterventions = new ArrayList<Intervention>();
274     String requete = "SELECT * FROM intervention;";
275     try {
276         uneBdd.seConnecter();
277         Statement unStat = uneBdd.getMaConnexion().createStatement();
278         ResultSet desResultats = unStat.executeQuery(requete);
279         while (desResultats.next()) {
280             Intervention unIntervention = new Intervention (
281                 desResultats.getInt("idintervention"),
282                 desResultats.getString("libelle"),
283                 desResultats.getString("dateintervention"),
284                 desResultats.getInt("idclient"),
285                 desResultats.getInt("idtechnicien")
286             );
287             lesInterventions.add(unIntervention);
288         }

```

```

2 import java.sql.ResultSet;
3 import java.sql.SQLException;
4 import java.sql.Statement;
5 import java.util.ArrayList;
6
7
8 import controleur.Technicien;
9 import controleur.CTI;
10 import controleur.Client;
11 import controleur.Intervention;
12
13 public class Modele {
14
15     private static Bdd uneBdd = new Bdd ("localhost", "orange", "root", "");
16
17     // INSERTION D'UN CLIENT
18     public static void insertClient (Client unClient) {
19         String requete = "INSERT INTO client VALUES (null, '"
20             + unClient.getNom() + "', '"
21             + unClient.getPrenom() + "', '"
22             + unClient.getEmail() + "', '"
23             + unClient.getTel() + "'"");";
24         try {
25             uneBdd.seConnecter();
26             Statement unStat = uneBdd.getMaConnexion().createStatement();
27             unStat.execute(requete);
28             unStat.close();
29             uneBdd.seDeconnecter();
30         } catch (SQLException exp) {
31             System.out.println("Erreur de requête : " + requete);
32         }

```

Modele :


```

153 // SELECTION DE TOUS LES TECHNICIENS
154 public static ArrayList<Technicien> selectAllTechniciens () {
155     ArrayList<Technicien> lesTechniciens = new ArrayList<Technicien>();
156     String requete = "SELECT * FROM technicien;";
157     try {
158         uneBdd.seConnecter();
159         Statement unStat = uneBdd.getMaConnexion().createStatement();
160         ResultSet desResultats = unStat.executeQuery(requete);
161         while (desResultats.next()) {
162             Technicien unTechnicien = new Technicien (
163                 desResultats.getInt("idtechnicien"),
164                 desResultats.getString("nom"),
165                 desResultats.getString("prenom"),
166                 desResultats.getString("qualification")
167             );
168             lesTechniciens.add(unTechnicien);
169         }
170         unStat.close();
171         uneBdd.seDeconnecter();
172     } catch (SQLException exp) {
173         System.out.println("Erreur de requête : " + requete);
174     }
175     return lesTechniciens;
176 }
177
178 // SUPPRESSION D'UN TECHNICIEN
179 public static void deleteTechnicien (int idtechnicien) {
180     String requete = "DELETE FROM technicien WHERE idtechnicien = " + idtechnicien;
181     try {
182         uneBdd.seConnecter();
183         Statement unStat = uneBdd.getMaConnexion().createStatement();
184         unStat.execute(requete);
185         unStat.close();
186         uneBdd.seDeconnecter();
187     } catch (SQLException exp) {
188         System.out.println("Erreur de requête : " + requete);
189     }
190 }
191

```

```

193 // SELECTION D'UN TECHNICIEN (SELECT WHERE)
194 public static Technicien selectWhereTechnicien (int idtechnicien) {
195     Technicien unTechnicien = null;
196     String requete = "SELECT * FROM technicien WHERE idtechnicien = "+idtechnicien+"";
197     try {
198         uneBdd.seConnecter();
199         Statement unStat = uneBdd.getMaConnexion().createStatement();
200         ResultSet unResultat = unStat.executeQuery(requete);
201         if (unResultat.next()) {
202             unTechnicien = new Technicien (
203                 unResultat.getInt("idtechnicien"),
204                 unResultat.getString("nom"),
205                 unResultat.getString("prenom"),
206                 unResultat.getString("qualification")
207             );
208         }
209         unStat.close();
210         uneBdd.seDeconnecter();
211     } catch (SQLException exp) {
212         System.out.println("Erreur de requête : " + requete);
213     }
214     return unTechnicien;
215 }
216
217 // EDITION D'UN TECHNICIEN
218 public static void updateTechnicien (Technicien unTechnicien) {
219     String requete = "UPDATE technicien SET nom = '"
220         + unTechnicien.getNom() + "', prenom = '"
221         + unTechnicien.getPrenom() + "', qualification = '"
222         + unTechnicien.getQualification() + "' WHERE idtechnicien = "+unTechnicien.getIdTechnicien()+"";
223     try {
224         uneBdd.seConnecter();
225         Statement unStat = uneBdd.getMaConnexion().createStatement();
226         unStat.execute(requete);
227         unStat.close();
228         uneBdd.seDeconnecter();
229     } catch (SQLException exp) {
230         System.out.println("Erreur de requête : " + requete);
231     }
232 }
233

```

```

234 // NOMBRE D'TECHNICIENS (COUNT())
235 public static int countTechniciens () {
236     int nbtechniciens = 0;
237     String requete = "SELECT count(*) as nb FROM technicien;";
238     try {
239         uneBdd.seConnecter();
240         Statement unStat = uneBdd.getMaConnexion().createStatement();
241         ResultSet unResultat = unStat.executeQuery(requete);
242         if (unResultat.next()) {
243             nbtechniciens = unResultat.getInt("nb");
244         }
245         unStat.close();
246         uneBdd.seDeconnecter();
247     } catch (SQLException exp) {
248         System.out.println("Erreur de requête : " + requete);
249     }
250     return nbtechniciens;
251 }
252
253 // INSERTION D'UNE INTERVENTION
254 public static void insertIntervention (Intervention unIntervention) {
255     String requete = "INSERT INTO intervention VALUES (null, '"
256         + unIntervention.getLibelle() + "', '"
257         + unIntervention.getDate() + "', '"
258         + unIntervention.getIdClient() + "', '"
259         + unIntervention.getIdTechnicien() + "')";
260     try {
261         uneBdd.seConnecter();
262         Statement unStat = uneBdd.getMaConnexion().createStatement();
263         unStat.execute(requete);
264         unStat.close();
265         uneBdd.seDeconnecter();
266     } catch (SQLException exp) {
267         System.out.println("Erreur de requête : " + requete);
268     }
269 }
270

```

```

270
271 // SELECTION DE TOUTE LES INTERVENTIONS
272 public static ArrayList<Intervention> selectAllInterventions () {
273     ArrayList<Intervention> lesInterventions = new ArrayList<Intervention>();
274     String requete = "SELECT * FROM intervention;";
275     try {
276         uneBdd.seConnecter();
277         Statement unStat = uneBdd.getMaConnexion().createStatement();
278         ResultSet desResultats = unStat.executeQuery(requete);
279         while (desResultats.next()) {
280             Intervention unIntervention = new Intervention (
281                 desResultats.getInt("idintervention"),
282                 desResultats.getString("libelle"),
283                 desResultats.getString("dateintervention"),
284                 desResultats.getInt("idclient"),
285                 desResultats.getInt("idtechnicien")
286             );
287             lesInterventions.add(unIntervention);
288         }
289         unStat.close();
290         uneBdd.seDeconnecter();
291     } catch (SQLException exp) {
292         System.out.println("Erreur de requête : " + requete);
293     }
294     return lesInterventions;
295 }
296
297 // SUPPRESSION D'UNE INTERVENTION
298 public static void deleteIntervention (int idintervention) {
299     String requete = "DELETE FROM intervention WHERE idintervention = " + idintervention;
300     try {
301         uneBdd.seConnecter();
302         Statement unStat = uneBdd.getMaConnexion().createStatement();
303         unStat.execute(requete);
304         unStat.close();
305         uneBdd.seDeconnecter();
306     } catch (SQLException exp) {
307         System.out.println("Erreur de requête : " + requete);
308     }
309 }

```

```

311 // SELECTION D'UNE INTERVENTION (SELECT WHERE)
312 public static Intervention selectWhereIntervention (int idintervention) {
313     Intervention unIntervention = null;
314     String requete = "SELECT * FROM intervention WHERE idintervention = "+idintervention+"";
315     try {
316         uneBdd.seConnecter();
317         Statement unStat = uneBdd.getMaConnexion().createStatement();
318         ResultSet unResultat = unStat.executeQuery(requete);
319         if (unResultat.next()) {
320             unIntervention = new Intervention (
321                 unResultat.getInt("idintervention"),
322                 unResultat.getString("libelle"),
323                 unResultat.getString("dateintervention"),
324                 unResultat.getInt("idclient"),
325                 unResultat.getInt("idtechnicien")
326             );
327         }
328         unStat.close();
329         uneBdd.seDeconnecter();
330     } catch (SQLException exp) {
331         System.out.println("Erreur de requête : " + requete);
332     }
333     return unIntervention;
334 }
335
336 // EDITION D'UNE INTERVENTION
337 public static void updateIntervention (Intervention unIntervention) {
338     String requete = "UPDATE intervention SET libelle = '"
339         + unIntervention.getLibelle() + "', dateintervention = '"
340         + unIntervention.getDate() + "', idclient = '"
341         + unIntervention.getIdClient() + "', idtechnicien = '"
342         + unIntervention.getIdTechnicien() + "' WHERE idtechnicien = "+unIntervention.getIdIntervention()+"";
343     try {
344         uneBdd.seConnecter();
345         Statement unStat = uneBdd.getMaConnexion().createStatement();
346         unStat.execute(requete);
347         unStat.close();
348         uneBdd.seDeconnecter();
349     } catch (SQLException exp) {
350         System.out.println("Erreur de requête : " + requete);
351     }
352 }

```

```

353
354 // NOMBRE D'INTERVENTIONS (COUNT())
355 public static int countInterventions () {
356     int nbinterventions = 0;
357     String requete = "SELECT count(*) as nb FROM intervention;";
358     try {
359         uneBdd.seConnecter();
360         Statement unStat = uneBdd.getMaConnexion().createStatement();
361         ResultSet unResultat = unStat.executeQuery(requete);
362         if (unResultat.next()) {
363             nbinterventions = unResultat.getInt("nb");
364         }
365         unStat.close();
366         uneBdd.seDeconnecter();
367     } catch (SQLException exp) {
368         System.out.println("Erreur de requête : " + requete);
369     }
370     return nbinterventions;
371 }
372
373 // SELECTION DE LA VUE : CTI
374 public static ArrayList<CTI> selectAllCTI () {
375     ArrayList<CTI> lesCTIs = new ArrayList<CTI>();
376     String requete = "SELECT * FROM cti;";
377     try {
378         uneBdd.seConnecter();
379         Statement unStat = uneBdd.getMaConnexion().createStatement();
380         ResultSet desResultats = unStat.executeQuery(requete);
381         while (desResultats.next()) {
382             CTI unCTI = new CTI (
383                 desResultats.getString("nomcli"),
384                 desResultats.getString("prenomcli "),
385                 desResultats.getString("nomtech"),
386                 desResultats.getString("prenomtech"),
387                 desResultats.getString("intervention"),
388                 desResultats.getString("dateintervention")
389             );
390             lesCTIs.add(unCTI);
391         }
392         unStat.close();
393         uneBdd.seDeconnecter();
394     } catch (SQLException exp) {
395         System.out.println("Erreur de requête : " + requete);
396     }
397     return lesCTIs;
398 }
399
400 }

```

Contrôleur

C_client.java

C_Technicien.java

C_Intervention.java

Client.java/Intervention.java/Technicien.java

Orange.java

CTI.java



Vue

Console.java

VueClient.java

VueCTI.java

VueTechnicien.java

VueIntervention.java