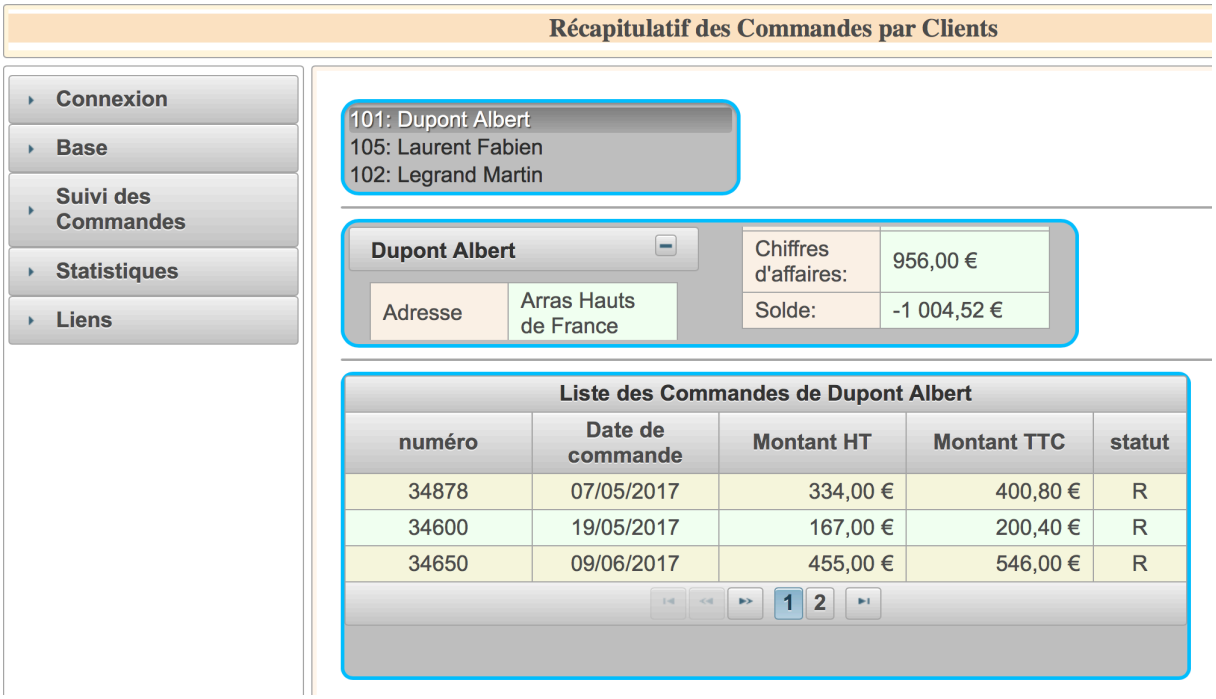
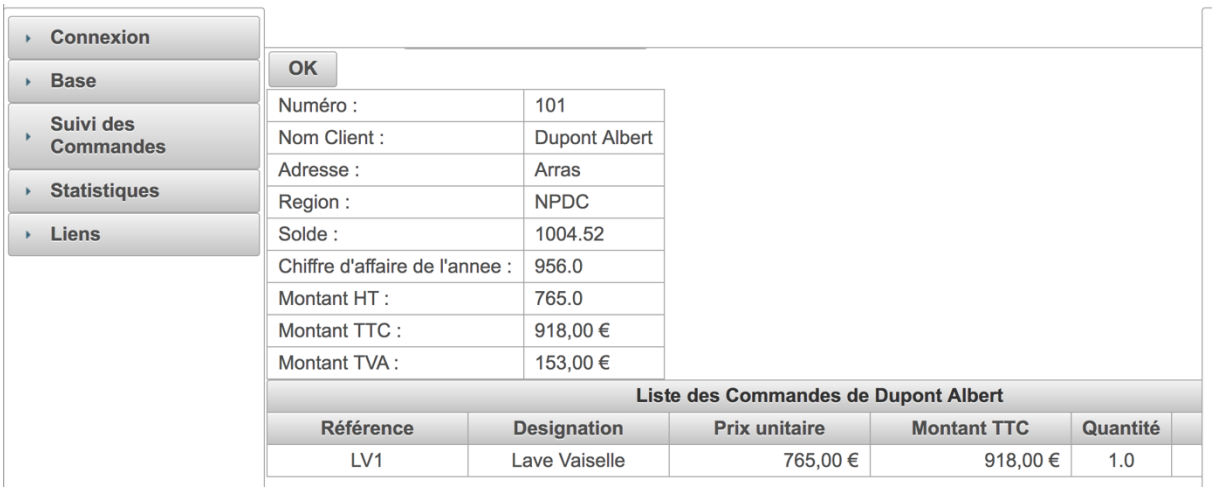
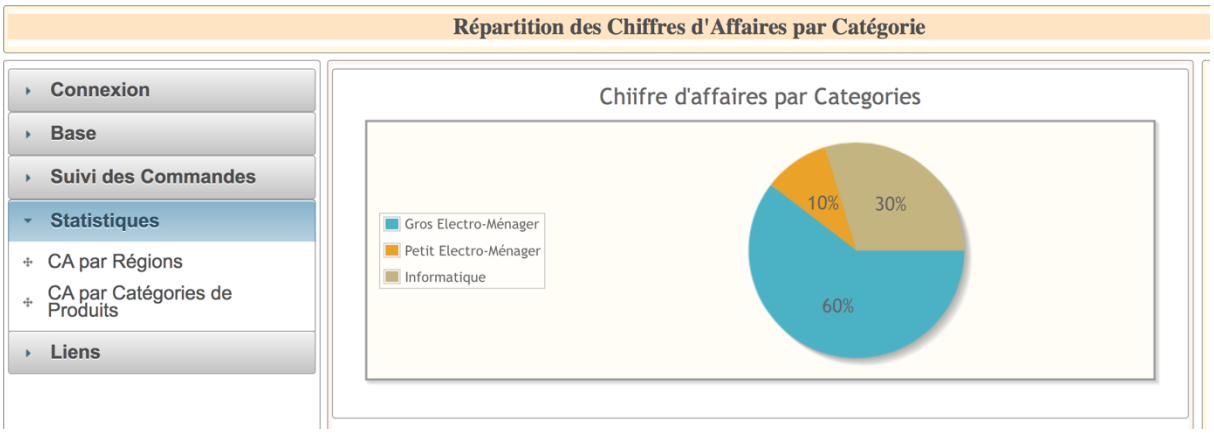
Documentation technique

1) Rappel de l'objectif Mission 1

1) Une page JSF permettant de consulter une commande à partir de la saisie de son numéro.
2) Une page JSF permettant d'afficher sous forme de camembert, le pourcentage de chiffre d'affaires pour l'année en cours que représente chaque catégorie de produit.
3) Intégrer les deux pages créées au menu de l'application et déployer l'application sur le serveur de production.
4) Compléter le diagramme de classes des entités fournies en faisant apparaître les méthodes métier.
5) Faire un diagramme de classes présentant les classes et interfaces des packages dao et bal.







Rappel de l'objectif Mission 3

Le but de cette première tache à réalisé lors de cette mission à été de développer un webservice REST proposant les ressources suivantes :

rest/resumecommande/{numerocommande}
rest/resumescommande/{numeroclient}
rest/resumeslignesdecommande/{numerocommande}
rest/resumeregion/{coderegion}
rest/resumesregion/tous

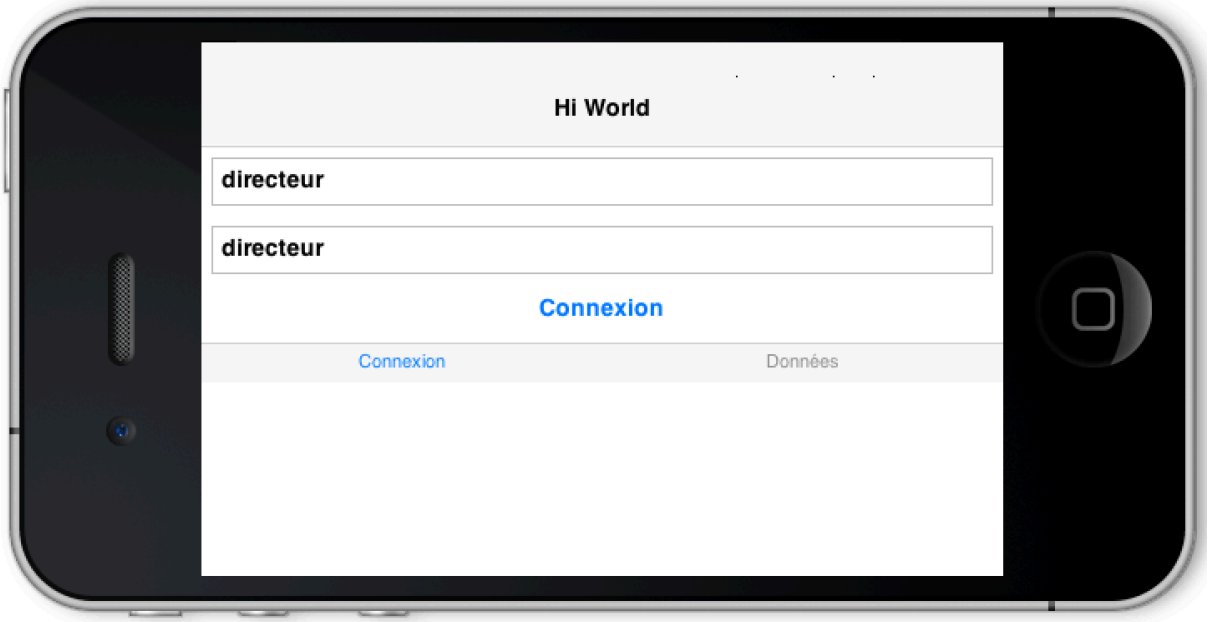
Avec les informations d'un resumecommande : numCommande numClient, nomClient, adrClient, coderegion, nomRegion etatCommande montantHT et montantTTC, totalTVA, remise totalNet (tva et remise comprises)

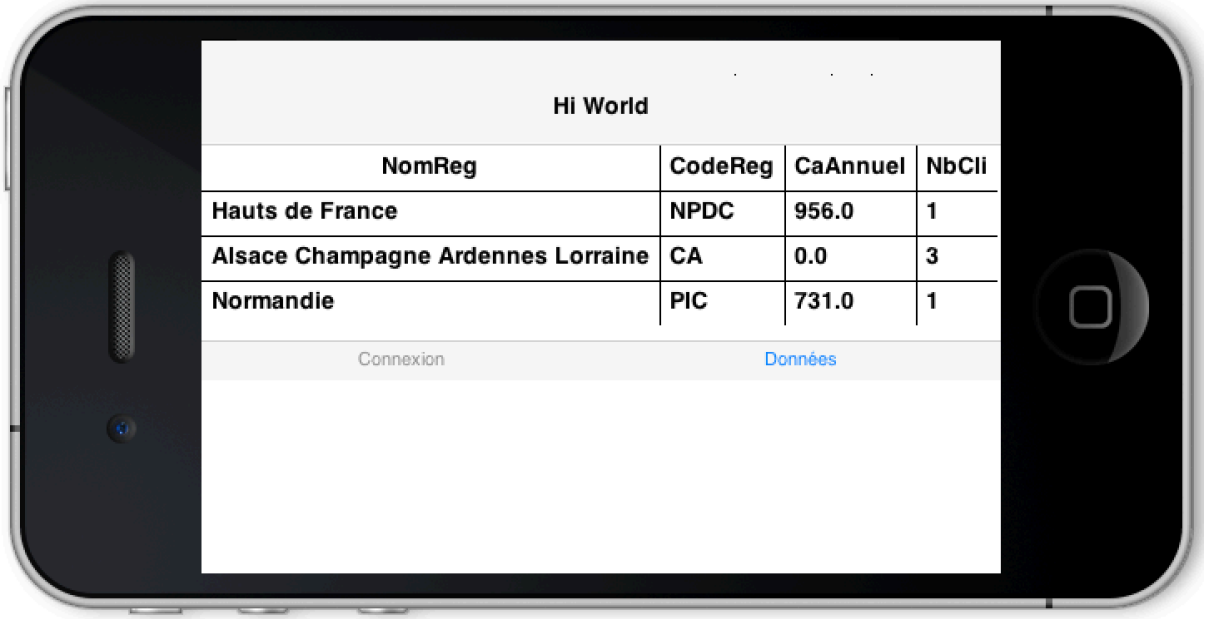
Les informations d'un resumelignedecommande : refProduit, desigPro, qteCommandee, prixUnitaire, montantHTLigne, montantTTCLigne.

Et les informations d'un resumeregion : coderegion, nomRegion, caAnnuelRegion, nombreclientsRegion.

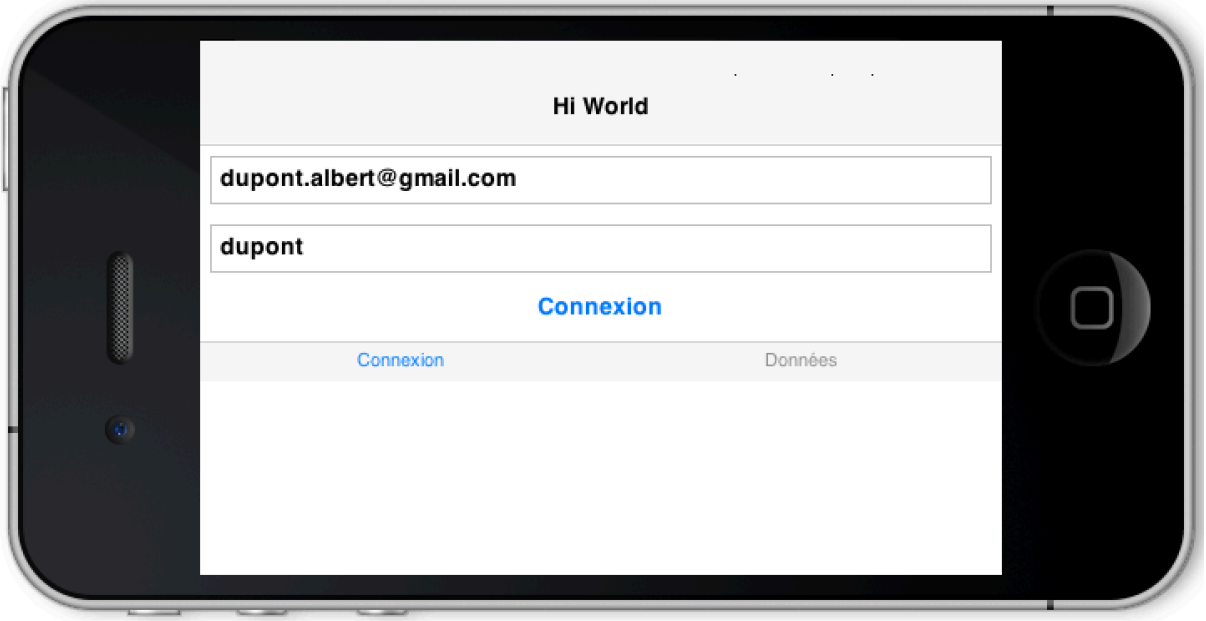
La deuxième tâches à été de développer deux applications mobiles déployables sur toute STA Mobile (Smartphone ou tablette) sous tous les systèmes (Android, OSX, ...):
Le framework utilisé pour développer ces applications multi-plateformes est CodeName One.

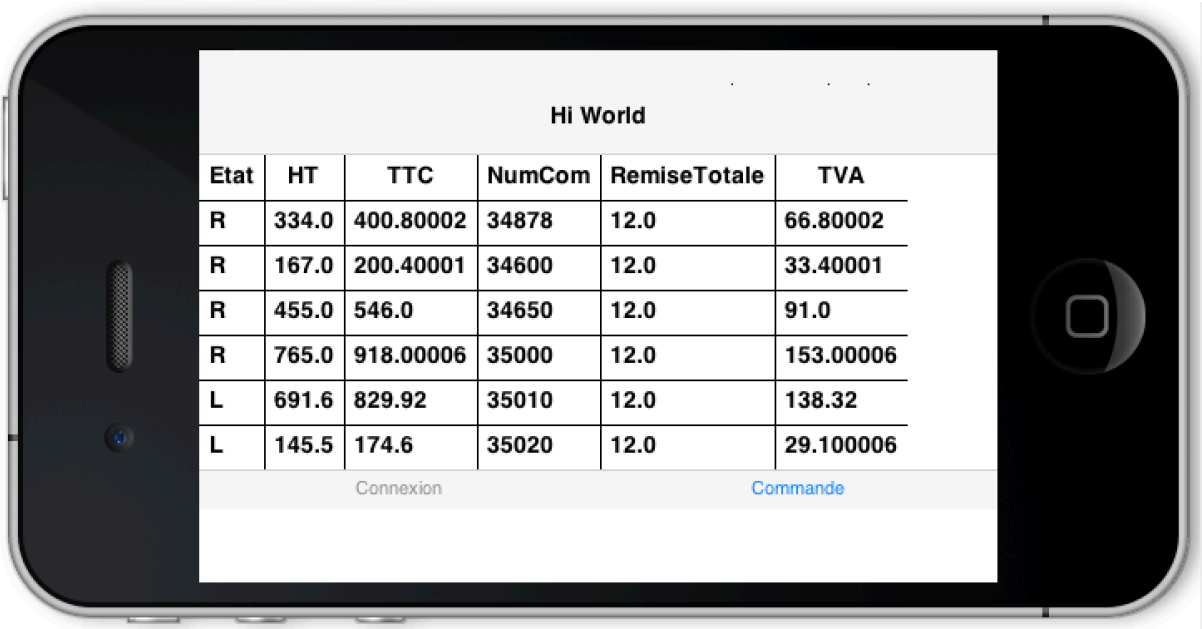
Application mobile destinée au directeur commercial.





Application mobile destinée aux clients.





Etat	HT	TTC	NumCom	RemiseTotale	TVA
R	334.0	400.80002	34878	12.0	66.80002
R	167.0	200.40001	34600	12.0	33.40001
R	455.0	546.0	34650	12.0	91.0
R	765.0	918.00006	35000	12.0	153.00006
L	691.6	829.92	35010	12.0	138.32
L	145.5	174.6	35020	12.0	29.100006

II) Diagramme de classe de l'application et Base de données Missions 1 et 3

Diagramme de classes des entités sans modifications

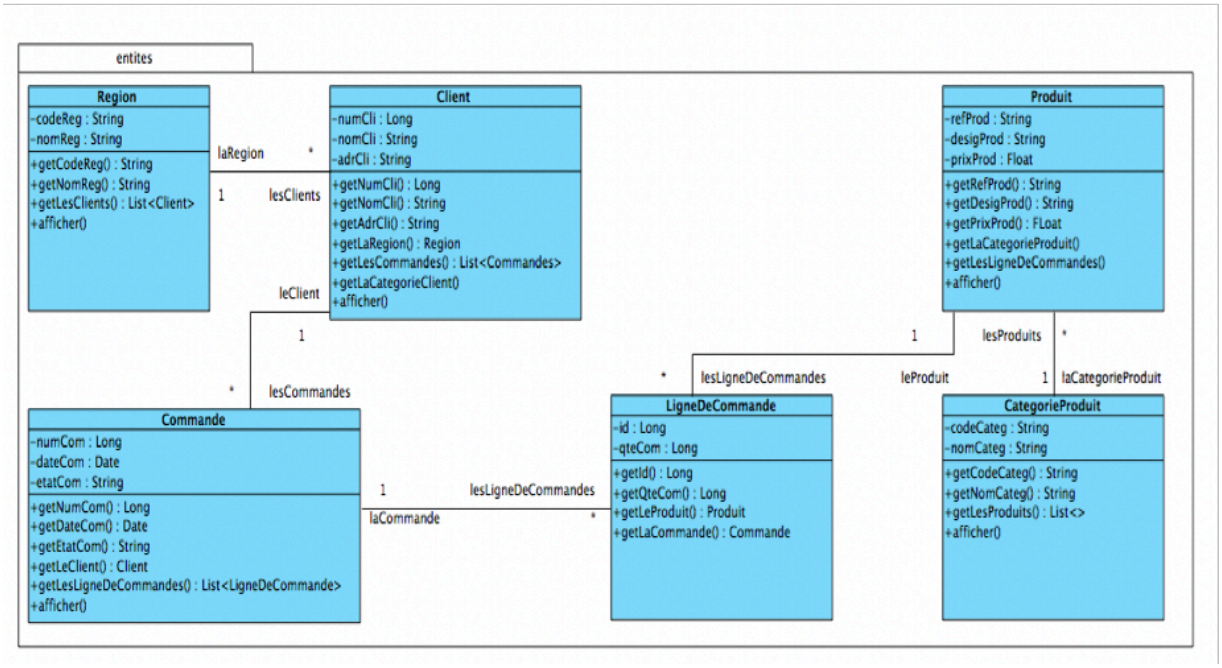
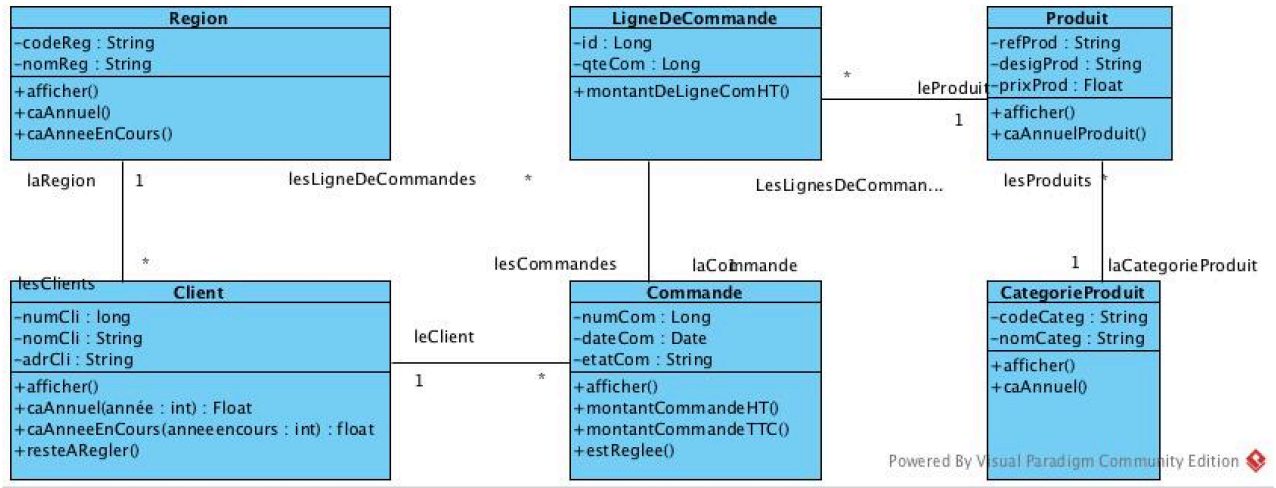
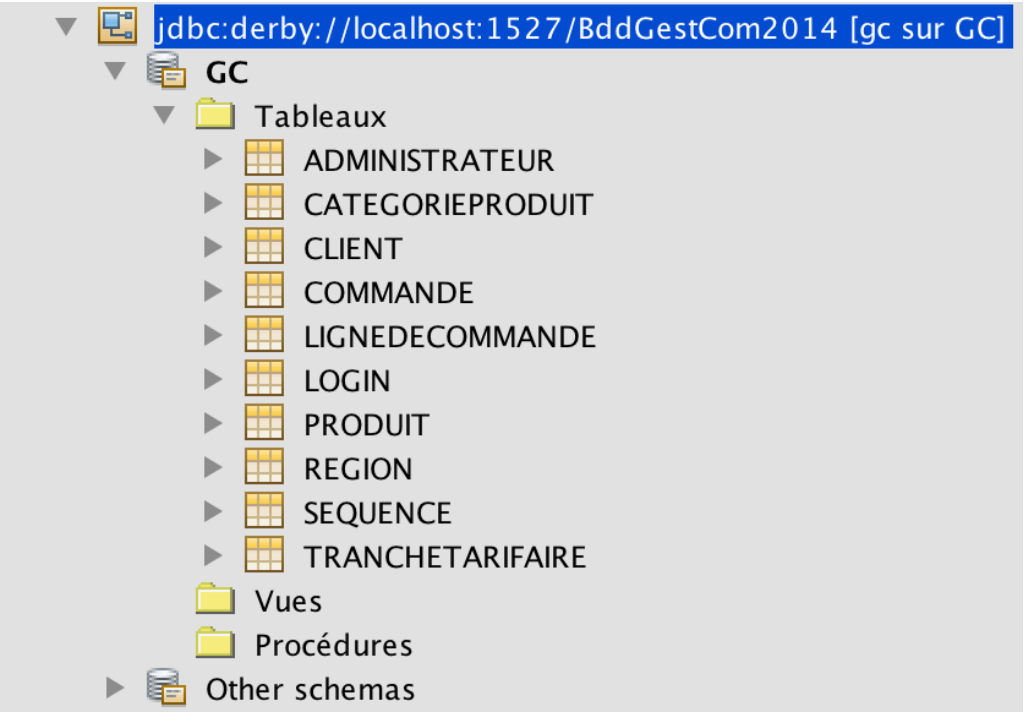
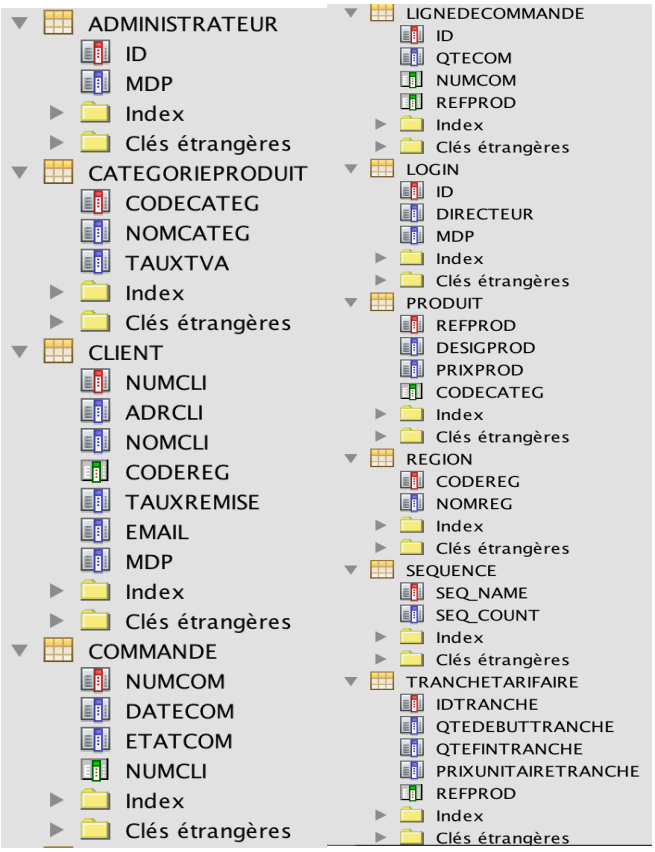


Diagramme de classes des entités après modifications



Base de donnée :





Ajout d'un champ email et mot de passe dans la table Client pour consulter ses commandes

Création d'un table Login avec un ID, un LOGIN et un MDP pour le directeur commercial

III) Intervention sur le code des applications

Entité Client

```java
@Entity
public class Client implements Serializable {

    @Id
    private Long   numCli;
    private String nomCli;
    private String adrCli;
    private Float tauxRemise;
    private String email;
    private String mdp;

    @JoinColumn(name="CODEREG")
    @ManyToOne
    private Region laRegion;

    @PrivateOwned
    @OneToMany(mappedBy = "leClient", cascade=CascadeType.ALL)
    private List<Commande> lesCommandes=new LinkedList();

    public Client() {}

    public Client(Long numCli, Float tauxRemise ,String nomCli, String adrCli, Region laRegion) {

        this.numCli = numCli;
        this.nomCli = nomCli;
        this.adrCli = adrCli;
        this.tauxRemise = tauxRemise;
        this.laRegion = laRegion;
    }



    @Override
    public String toString() {return nomCli ;}

    public void  afficher(){System.out.printf("%4d %-20s %-60s %10.2",numCli,nomCli,adrCli,tauxRemise);}
```

```java
//<editor-fold defaultstate="collapsed" desc="METHODES METIERS">
public Float caAnnuel(int pAnnee)            {

    Float ca=0f;
    for (Commande cmd : lesCommandes){
        if(cmd.getEtatCom().equalsIgnoreCase("R") && dansAnnee(cmd.getDateCom(), pAnnee)){
            ca+=cmd.montantCommandeHT();
        }
    }
    return ca;
}

public Float caAnneeEnCours()                { return this.caAnnuel(anneeCourante()); }

public Float caMoisEnCours(Client pClient)   {
    return this.caAnnuel(anneeCourante(), moisCourant());
}

public Float caAnnuel(int pAnnee, int pMois) {
    Float ca=0f;
    for (Commande cmd : lesCommandes){

        if( cmd.getEtatCom().equalsIgnoreCase("R")&&
                dansAnneeEtMois(cmd.getDateCom(), pAnnee, pMois)
                )
        {
            ca+=cmd.montantCommandeHT();
        }
    }
    return ca;
}

public Float resteARegler()                  {
    Float valeur=0F;
    for ( Commande cmd : lesCommandes) {
        if (! cmd.estReglee() ){
            valeur+= cmd.montantCommandeTTC();
        }
    }
    return valeur;
```

Entité Région

```
@Entity
public class Region implements Serializable {

    @Id
    private String codeReg;
    private String nomReg;

    @PrivateOwned
    @OneToMany(mappedBy = "laRegion", cascade=CascadeType.ALL)
    private List<Client> lesClients=new LinkedList<Client>();

    public Region() {}

    public Region(String codeRegion, String nomRegion) {

        this.codeReg = codeRegion;
        this.nomReg  = nomRegion;
    }

    public void afficher(){ System.out.printf("%-10s %20s",codeReg,nomReg);}



    public Float caAnnuel(int pAnnee) {

        Float ca=0f;

        for (Client cli : lesClients) {

            for (Commande cmd : cli.getLesCommandes()){

              if( dansAnnee(cmd.getDateCom(), pAnnee)&& cmd.estReglee()){

                  ca+=cmd.montantCommandeHT();
              }
            }
        }
        return ca;
    }
```

Entité CatégorieProduit

```java
@Entity
public class CategorieProduit implements Serializable {

    @Id
    private String codeCateg;
    private String nomCateg;
    private Float tauxTva;
    @PrivateOwned
    @OneToMany(mappedBy = "laCategorie", cascade=CascadeType.ALL)
    private List<Produit> lesProduits= new LinkedList();

    public CategorieProduit() {}

    public CategorieProduit(String codeCateg, String nomCateg, Float tauxTva) {
        this.codeCateg = codeCateg;
        this.nomCateg = nomCateg;
        this.tauxTva = tauxTva;
    }

    public void afficher(){

        System.out.printf("%-8s %-20s %10.2f",codeCateg,nomCateg,tauxTva);
    }

    public Float caAnnuel(){

        Float ca=0f;
        for(Produit produit: lesProduits){
            ca+=produit.caAnnuelProduit(utilitaires.UtilDate.anneeCourante());
        }
        return ca;
    }

    GETTERS ET SETTERS
```

Entité Commande

```
15  @Entity
16  public class Commande implements Serializable {
17
18      @Id
19      private Long numCom;
20
21      @Temporal(javax.persistence.TemporalType.DATE)
22      private Date dateCom;
23
24      private String etatCom;
25
26      @PrivateOwned
27      @OneToMany(mappedBy = "laCommande", cascade=CascadeType.ALL)
28      private List<LigneDeCommande> lesLignesDeCommande=new LinkedList();
29
30      @JoinColumn(name="NUMCLI")
31      @ManyToOne
32      private Client leClient;
33
34      public Commande(){}
35
36      public Commande(Long numCom,Date dateCom, String etatCom, Client leClient) {
37
38          this.numCom   = numCom;
39          this.dateCom  = dateCom;
40          this.etatCom  = etatCom;
41          this.leClient = leClient;
42      }
43
44      public void afficher(){
45
46          System.out.printf("%6d %-10s %2s", numCom,dateCom,etatCom);
47      }
48
49      METHODES METIERS
49      //<editor-fold defaultstate="collapsed" desc="METHODES METIERS">
50
51      public Float montantCommandeHT()  {
52
53          Float montant=0F;
54          for(LigneDeCommande lgdc: lesLignesDeCommande){
55
56              montant+=lgdc.montantDeLigneComHT();
57          }
58
59          return montant;
60      }
61
62      public Float montantCommandeTTC() {
63
64          return montantCommandeHT()*(1+0.2F);
65      }
66
67      public boolean estReglee()        {
68          return this.etatCom.equalsIgnoreCase("R");
69      }
70
71      //</editor-fold>
72
73      GETTERS ET SETTERS
105 }
```

Entité Directeur

```java
package entites;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

/**
 *
 * @author QuentinVarlet
 */
@Entity
@Table(name="Administrateur")

public class Directeur implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id

    private String id;

    private String mdp;

    public Directeur() {
    }



    Getters & Setters
```

Entité LigneDeCommande

```java
@Entity
public class LigneDeCommande implements Serializable {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long id;

    private Float qteCom;

    @JoinColumn(name="NUMCOM")
    @ManyToOne
    private Commande laCommande;

    @JoinColumn(name="REFPROD")
    @ManyToOne
    private Produit leProduit;

    public LigneDeCommande(){}

    public LigneDeCommande(Commande laCommande, Produit leProduit, Float qteCom ) {

        this.qteCom = qteCom;
        this.laCommande = laCommande;
        this.leProduit = leProduit;
    }

    public Float montantDeLigneComHT(){



        return qteCom*leProduit.getPrixProd();
    }

    GETTERS ET SETTERS
}
```

Entité Produit

```java
15  @Entity
16  public class Produit implements Serializable {
17
18      @Id
19      private String refProd;
20
21      private String desigProd;
22      private Float  prixProd;
23
24      @OneToMany(mappedBy = "leProduit")
25      private List<TrancheTarifaire>   lesTranches=new LinkedList();
26
27      @PrivateOwned
28      @OneToMany(mappedBy = "leProduit", cascade=CascadeType.ALL)
29      private List<LigneDeCommande>   lesLignesDeCommande=new LinkedList();
30
31      @JoinColumn(name="CODECATEG")
32      @ManyToOne
33      private CategorieProduit laCategorie;
34
35      public Produit (){ }
36
37      public Produit(String refProd,String desigProd, Float prixProd, CategorieProduit categProd) {
38
39          this.refProd     = refProd;
40          this.desigProd   = desigProd;
41          this.prixProd    = prixProd;
42          this.laCategorie = categProd;
43      }
44
45      public void afficher(){
46
47          System.out.printf("%-10s  %-20s  %5.2f",refProd,desigProd,prixProd);
48       }
49
50      // Méthodes métiers
```

```java
50      // Méthodes métiers
51
52      public Float caAnnuelProduit(int pAnnee) {
53
54          Float ca=0F;
55
56          for (LigneDeCommande lgdc: lesLignesDeCommande){
57
58            Commande cmd     = lgdc.getLaCommande();
59            Date     dateCmd = cmd.getDateCom();
60            String   etatCmd = cmd.getEtatCom();
61
62            if( annee(dateCmd)==pAnnee && etatCmd.equalsIgnoreCase("R") ){
63
64                ca+=lgdc.getQteCom()*lgdc.getLeProduit().getPrixProd();
65            }
66          }
67          return ca;
68       }
69
70      public Float caAnnuelProduit(int pAnnee, int pMois) {
71
72          Float ca=0F;
73
74          for (LigneDeCommande lgdc: lesLignesDeCommande){
75
76            Commande cmd     = lgdc.getLaCommande();
77            Date     dateCmd = cmd.getDateCom();
78            String   etatCmd = cmd.getEtatCom();
79
80            if( dansAnneeEtMois(dateCmd, pAnnee, pMois) && etatCmd.equalsIgnoreCase("R")){
81
82                ca+=lgdc.getQteCom()*lgdc.getLeProduit().getPrixProd();
83            }
84          }
85          return ca;
86       }
```

Les différentes DAO

DAO Catégorie

```java
package dao.consultation.categorie;

import entites.CategorieProduit;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;


public class DaoCategorieImpl implements DaoCategorie {

@PersistenceContext private EntityManager em;

    @Override
    public List<CategorieProduit> getToutesLesCategoriesProduits() {
        return em.createQuery("Select c from CategorieProduit c").getResultList();
    }
}
```

DAO Client

```java
package dao.consultation.client;
import entites.Client;
import java.io.Serializable;
import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

@Stateless
public class DaoClientImpl  implements DaoClient, Serializable {

    @PersistenceContext private EntityManager em;

    @Override
    public Client getLeClient(Long pNumcli) {

        return em.find(Client.class, pNumcli);
    }

    @Override
    public Client getLeClientMail(String email){
        Client c = null;
        try{
            c= (Client) em.createQuery("Select c From Client c where c.email = '"+ email +"'").getSingleResult();
        }
        catch(Exception ex){}
        return c;
    }

    @Override
    public List<Client> getTousLesClients() {

        return em.createQuery("Select c From Client c order by c.nomCli").getResultList();
    }
}
```

DAO Commande

```java
package dao.consultation.commande;
import entites.Commande;
import java.io.Serializable;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

public class DaoCommandeImpl   implements DaoCommande, Serializable{

    @PersistenceContext private EntityManager em;

    @Override
    public Commande getCommande(Long pNumCom) {
        return em.find(Commande.class, pNumCom);
    }

    @Override
    public List<Commande> getToutesLesCommandes() {
        return em.createQuery("Select c from Commande c").getResultList();
    }


    @Override
    public List<Commande> getLesCommandes( int pAnnee) {

       String reqJPQL = "Select com "+
                        "From   Commande com "+
                        "Where  FUNC('YEAR',com.dateCom)= :pA";


        Query qryListeCom= em.createQuery(reqJPQL);

        qryListeCom.setParameter("pA", pAnnee);

        return qryListeCom.getResultList();
    }

    @Override
    public List<Commande> getLesCommandes( int pAnnee, String pEtat) {

       String reqJPQL = "Select com "+
                        "From   Commande com "+
                        "Where  FUNC('YEAR',com.dateCom)= :pA " +
                        "And    com.etatCom=:pE";
        Query qryListeCom= em.createQuery(reqJPQL);

        qryListeCom.setParameter("pA", pAnnee);
        qryListeCom.setParameter("pE", pEtat);

        return qryListeCom.getResultList();
    }

    @Override
    public List<Commande> getLesCommandes( int pAnnee, int pMois) {

       String reqJPQL = "Select com "+
                        "From   Commande com "+
                        "Where  FUNC('YEAR' ,com.dateCom)= :pA " +
                        "And    FUNC('MONTH',com.dateCom)= :pM"
                         ;
       Query qryListeCom= em.createQuery(reqJPQL);

       qryListeCom.setParameter("pA", pAnnee);
       qryListeCom.setParameter("pM", pMois);

       return qryListeCom.getResultList();
    }

    public List<Commande> getLesCommandes( int pAnnee, int pMois, String pEtat) {

       String reqJPQL = "Select com "+
                        "From   Commande com "+
                        "Where  FUNC('YEAR' ,com.dateCom)= :pA " +
                        "And    FUNC('MONTH',com.dateCom)= :pM " +
                        "And    com.etatCom=:pE"  ;

       Query qryListeCom= em.createQuery(reqJPQL);

       qryListeCom.setParameter("pA", pAnnee);
       qryListeCom.setParameter("pM", pMois);
       qryListeCom.setParameter("pE", pEtat);

       return qryListeCom.getResultList();
    }
}
```

DAO Directeur

```java
package dao.consultation.directeur;

import entites.Directeur;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

/**
 *
 * @author QuentinVarlet
 */
@Stateless

public class DaoDirecteur {

        @PersistenceContext private EntityManager em;


    public Directeur getLeDirecteur(String pNumDir) {

        return em.find(Directeur.class, pNumDir);
    }

}
```

DAO Produit

```java
package dao.consultation.produit;
import entites.Produit;
import java.io.Serializable;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

@Stateless
public class DaoProduitImpl  implements DaoProduit, Serializable{

    @PersistenceContext private EntityManager em;

    @Override
    public Produit getProduit(String pRefProd) {
        return em.find(Produit.class, pRefProd);
    }
}

```

Dao Région

```java
@Stateless
public class DaoRegionImpl  implements DaoRegion,Serializable {

    @PersistenceContext private EntityManager em;

    @Override
    public Region getRegion(String pCodeReg) {
            return em.find(Region.class, pCodeReg);
    }

    @Override
    public List<Region> getToutesLesRegions() {

        return em.createQuery("Select r from Region r").getResultList();
    }

}

```

Le WEBSERVICES

```java
package web.services.rest;

import dao.consultation.client.DaoClient;
import dao.consultation.commande.DaoCommandeImpl;
import dao.consultation.directeur.DaoDirecteur;
import entites.Directeur;
import dao.consultation.region.DaoRegion;
import dto.DtoCommande;
import dto.DtoLigneCommande;
import dto.DtoRegion;
import entites.Client;
import entites.Commande;
import entites.LigneDeCommande;
import entites.Region;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.Stateless;
import javax.inject.Inject;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;

/**
 *
 * @author QuentinVarlet
 */
@Stateless
@Path ("gc")
public class WebServices {


    @Inject DaoCommandeImpl dao;

    @GET
    @Path("resumecommande/{numcommande}")
    @Produces({"application/xml","application/json"})

    public DtoCommande getCommande(@PathParam("numcommande")Long numCom) {

        Commande cmd = dao.getCommande(numCom);
        DtoCommande dto = new DtoCommande();


        dto.setNumCom(cmd.getNumCom());
        dto.setNumClient(cmd.getLeClient().getNumCli());
        dto.setNomClient(cmd.getLeClient().getNomCli());
        dto.setAdrClient(cmd.getLeClient().getAdrCli());
        dto.setCodeRegion(cmd.getLeClient().getLaRegion().getCodeRegion());
        dto.setNomRegion(cmd.getLeClient().getLaRegion().getNomRegion());
        dto.setEtatCom(cmd.getEtatCom());
        dto.setMontantHT(cmd.montantCommandeHT());
        dto.setMontantTTC(cmd.montantCommandeTTC());
        dto.setTotalTVA(cmd.montantCommandeTTC()-cmd.montantCommandeHT());
        dto.setRemiseTotalNet(cmd.getLeClient().getTauxRemise());

        return dto;

    }

    @Inject DaoClient daoClient;

    @GET
    @Path("resumescommande/{numeroclient}")
    @Produces({"application/xml","application/json"})
    public List<DtoCommande> getListDtoCommande(@PathParam("numeroclient")Long numCli) {

        Client cli = daoClient.getLeClient(numCli);

        List<DtoCommande> listeDtoCom= new LinkedList();

        for (Commande c : cli.getLesCommandes()){

            DtoCommande dto = new DtoCommande();

            dto.setNumCom(c.getNumCom());
            dto.setNumClient(c.getLeClient().getNumCli());
            dto.setNomClient(c.getLeClient().getNomCli());
            dto.setAdrClient(c.getLeClient().getAdrCli());
            dto.setCodeRegion(c.getLeClient().getLaRegion().getCodeRegion());
            dto.setNomRegion(c.getLeClient().getLaRegion().getNomRegion());
            dto.setEtatCom(c.getEtatCom());
            dto.setMontantHT(c.montantCommandeHT());
            dto.setMontantTTC(c.montantCommandeTTC());
            dto.setTotalTVA(c.montantCommandeTTC()-c.montantCommandeHT());
            dto.setRemiseTotalNet(c.getLeClient().getTauxRemise());

            listeDtoCom.add(dto);

        }

        return listeDtoCom;

    }
```

```java
    @Inject DaoCommandeImpl daol;

    @GET
    @Path("resumeslignedecommande/{numerocommande}")
    @Produces({"application/xml","application/json"})
    public List<DtoLigneCommande> getListDtoLigneCommande(@PathParam("numerocommande")Long numCom) {

        Commande com = daol.getCommande(numCom);

        List<DtoLigneCommande> listeDtoLigneCom= new LinkedList();

        for (LigneDeCommande lignecom : com.getLesLignesDeCommande()){

            DtoLigneCommande dto = new DtoLigneCommande();

            dto.setRefProduit(lignecom.getLeProduit().getRefProd());
            dto.setDesigPro(lignecom.getLeProduit().getDesigProd());
            dto.setQteCommande(lignecom.getQteCom());
            dto.setMontantHTLigne(lignecom.montantDeLigneComHT());
            dto.setMontantTTCLigne(lignecom.montantDeLigneComHT()*1.2f);
            dto.setPrixUnitaire(lignecom.getLeProduit().getPrixProd());

            listeDtoLigneCom.add(dto);

        }

        return listeDtoLigneCom;

    }

    @Inject DaoRegion daoreg;

    @GET
    @Path("resumeregion/{coderegion}")
    @Produces({"application/xml","application/json"})

    public DtoRegion getRegion(@PathParam("coderegion")String codeReg) {

        Region reg = daoreg.getRegion(codeReg);
        DtoRegion dto = new DtoRegion();

        dto.setCodeRegion(codeReg);
        dto.setNomRegion(reg.getNomRegion());
        dto.setCaAnnuelRegion(reg.caAnneeEnCours());
        dto.setNombreClientsRegions(reg.nombreClientsRegion());

        return dto;

    }

    @GET
    @Path("resumesregion/tous")
    @Produces({"application/xml","application/json"})
    public List<DtoRegion> getListDtoRegion() {

        List<DtoRegion> listeDtoReg= new LinkedList();

        for (Region reg : daoreg.getToutesLesRegions()){

            DtoRegion dto = new DtoRegion();

            dto.setCodeRegion(reg.getCodeRegion());
            dto.setNomRegion(reg.getNomRegion());
            dto.setCaAnnuelRegion(reg.caAnneeEnCours());
            dto.setNombreClientsRegions(reg.nombreClientsRegion());
            listeDtoReg.add(dto);
        }
        return listeDtoReg;
    }

    @Inject DaoDirecteur daoDirecteur;


    @GET
    @Path("login/{id}/{mdp}")
    @Produces({"text/plain"})

    public String login(@PathParam("id")String id, @PathParam("mdp") String mdp){

        //String mdp_saisie = null;
        Directeur d = daoDirecteur.getLeDirecteur(id);
        if(d!= null && d.getMdp().equals(mdp) ){
            return "OK";
        }
        return "KO";
    }

    @GET
    @Path("client/{email}/{mdp}")
    @Produces({"text/plain"})

    public String client(@PathParam("email")String email, @PathParam("mdp") String mdp){

        //String mdp_saisie = null;
        Client c = daoClient.getLeClientMail(email);
        if(c!= null && c.getMdp().equals(mdp) ){
            return c.getNumCli().toString();
        }
        return "KO";
    }
}
```

Les différents DTO

DTO Commande

```
6   package dto;
7
8   import javax.xml.bind.annotation.XmlRootElement;
9
10  @XmlRootElement
11  public class DtoCommande {
12
13      private Long numClient;
14      private Long numCom;
15      private String etatCom;
16      private String nomClient;
17      private String adrClient;
18      private String codeRegion;
19      private String nomRegion;
20      private Float montantHT;
21      private Float montantTTC;
22      private Float totalTVA;
23      private Float remiseTotalNet;
24
25      Getters & Setters
114
115 }
```

DTO LigneCommande

```
6   package dto;
7
8   import javax.xml.bind.annotation.XmlRootElement;
9
10  @XmlRootElement
11  public class DtoLigneCommande {
12
13      private String refProduit;
14      private String desigPro;
15      private Float qteCommande;
16      private Float prixUnitaire;
17      private Float montantHTLigne;
18      private Float montantTTCLigne;
19
20      Getters & Setters
69
70
71  }
```

DTO Region

```
6   package dto;
7
8   import javax.xml.bind.annotation.XmlRootElement;
9
10  @XmlRootElement
11  public class DtoRegion {
12
13      private String codeRegion;
14      private String nomRegion;
15      private Float CaAnnuelRegion;
16      private int nombreClientsRegions;
17
18      Getters & Setters
51
52
53  }
54
```

Application mobile destinée au directeur commercial.

Classe DirecteurCommercial

```java
package com.mycompany.directeur;
import com.codename1.ui.Display;
import com.codename1.ui.Form;
import com.codename1.ui.Dialog;
import userclasses.StateMachine;
import com.codename1.io.Log;
import com.codename1.ui.Toolbar;
/**
 * This file was generated by <a href="https://www.codenam
 * of building native mobile applications using Java.
 */
public class DirecteurCommercial {

    private Form current;

    public void init(Object context) {
        Toolbar.setGlobalToolbar(true);
    }

    public void start() {
        if(current != null){
            current.show();
            return;
        }
        new StateMachine("/theme");
    }

    public void stop() {
        current = Display.getInstance().getCurrent();
        if(current instanceof Dialog) {
            ((Dialog)current).dispose();
            current = Display.getInstance().getCurrent();
        }
    }

    public void destroy() {
    }
}
```

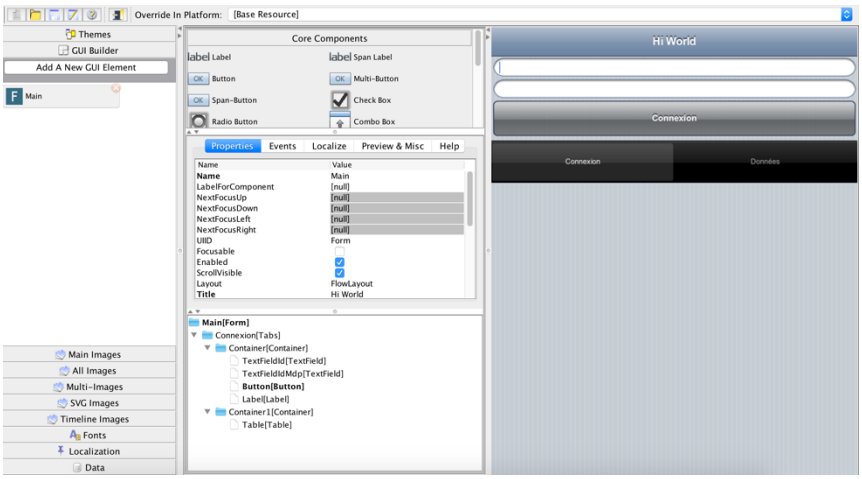
La classe ParametresReseau

```java

package userclasses.util.paramreseau;

public class ParametresReseau {

  private static String ADRESSESERVEUR      = "192.168.43.116";     // DISTANTE OVH
 // private static String ADRESSESERVEUR      = "172.16.21.27";  // LOCALE   LYCEE
 // private static String ADRESSESERVEUR      = "192.168.1.3";     // LOCALE   AT HOME


    private static String PORT                = "8080";
    private static String APPLI               = "Web_Appli_GC_PPES";
    private static String ENTREE_SERVICE      = "rest";

    public  static String RACINE_SERVICE_REST = "http://"+
                                                ADRESSESERVEUR+":"+
                                                PORT+"/"+
                                                APPLI+"/"+
                                                ENTREE_SERVICE+"/";
}
```

Classe StateMachine

```java
8   package userclasses;
9
10  import generated.StateMachineBase;
11  import com.codename1.ui.*;
12  import com.codename1.ui.events.*;
13  import com.codename1.ui.table.Table;
14  import com.codename1.xml.Element;
15  import java.util.Vector;
16  import userclasses.util.modele.ModeleDeTableau;
17  import static userclasses.util.parser.ParserArbreXML.getLesElementsFils;
18  import static userclasses.util.parser.ParserArbreXML.getValeur;
19  import userclasses.util.parser.RequetePourArbreXML;
20  import userclasses.util.parser.RequetePourTexteBrut;
21
22  public class StateMachine extends StateMachineBase {
23
24      private RequetePourTexteBrut requeteRestConnexion;
25      private RequetePourArbreXML requeteRestRegion;
26      private Element eltRegion;
27      private String cx;
28      private Table tableau;
29      private ModeleDeTableau modeleDuTableau;
30
31      public StateMachine(String resFile) {
32          super(resFile);
33
34          // do not modify, write code in initVars and initialize class members there,
35          // the constructor might be invoked too late due to race conditions that might o
36      }
37
38      @Override
39      protected void postMain(Form f) {
40          super.postMain(f); //To change body of generated methods, choose Tools | Templat
41
42          requeteRestConnexion = new RequetePourTexteBrut();
43          requeteRestRegion = new RequetePourArbreXML();
44          //requeteRestExploitation.executer("gc/login/{id}{mdp}");
45          modeleDuTableau   = new ModeleDeTableau("NomReg","CodeReg","CaAnnuel", "NbCli");
46          tableau = this.findTable();
47      }
49      @Override
50      protected void onMain_ButtonAction(Component c, ActionEvent event){
51          super.onMain_ButtonAction(c, event);
52
53          String id = this.findTextFieldId().getText();
54          String mdp = this.findTextFieldIdMdp().getText();
55
56          if (!id.equals("") && !mdp.equals("")){
57              requeteRestConnexion.executer("gc/login/"+id+"/"+mdp);
58              cx = requeteRestConnexion.getTexte();
59              System.out.println(cx);
60              if(cx.equals("OK")){
61
62                  requeteRestRegion.executer("gc/resumesregion/tous");
63                  this.eltRegion = requeteRestRegion.getRacine();
64                   System.out.println(eltRegion);
65                   Vector<Element>  lesElementsRegion  = getLesElementsFils(eltRegion,"dtoregion");
66                   tableau.setVisible(true);
67                   //System.out.println(lesElementsParcelles);
68                   for ( Element elementReg: lesElementsRegion){
69                       // System.out.println(eltRegion);
70                        modeleDuTableau.ajouterRangee(
71                        getValeur(elementReg,"nomregion"),
72                        getValeur(elementReg,"coderegion"),
73                        getValeur(elementReg,"caannuelregion"),
74                        getValeur(elementReg,"nombreclientsregions")
75                        );
76                   }
77                 tableau.setModel(modeleDuTableau);
78              }
79          }
80      }
81  }
```

Le fichier thème



Application mobile destinée aux clients

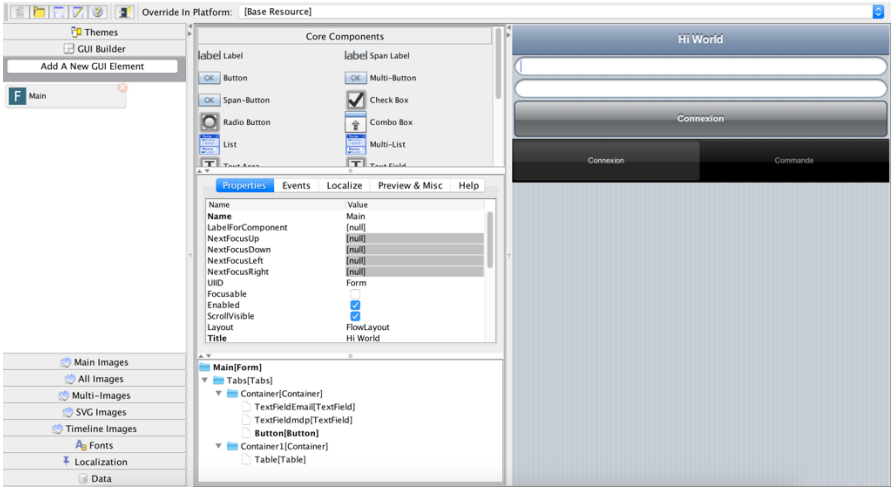
Classe MyApplication (Client)

```java
package com.mycompany.client;

import com.codename1.ui.Display;
import com.codename1.ui.Form;
import com.codename1.ui.Dialog;
import userclasses.StateMachine;
import com.codename1.io.Log;
import com.codename1.ui.Toolbar;

public class MyApplication {

    private Form current;

    public void init(Object context) {
        // Enable Toolbar on all Forms by default
        Toolbar.setGlobalToolbar(true);

    }

    public void start() {
        if(current != null){
            current.show();
            return;
        }
        new StateMachine("/theme");
    }

    public void stop() {
        current = Display.getInstance().getCurrent();
        if(current instanceof Dialog) {
            ((Dialog)current).dispose();
            current = Display.getInstance().getCurrent();
        }
    }

    public void destroy() {
    }
}
```

Classe ParametreReseau

```java

package userclasses.util.paramreseau;

public class ParametresReseau {

  private static String ADRESSESERVEUR      = "192.168.43.116";     // DISTANTE OVH
 // private static String ADRESSESERVEUR      = "172.16.21.27";  // LOCALE   LYCEE
 // private static String ADRESSESERVEUR      = "192.168.1.3";     // LOCALE   AT HOME


    private static String PORT                = "8080";
    private static String APPLI               = "Web_Appli_GC_PPES";
    private static String ENTREE_SERVICE      = "rest";

    public  static String RACINE_SERVICE_REST = "http://"+
                                                ADRESSESERVEUR+":"+
                                                PORT+"/"+
                                                APPLI+"/"+
                                                ENTREE_SERVICE+"/";
}
```

Classe StateMachine

```java
8   package userclasses;
9
10  import generated.StateMachineBase;
11  import com.codename1.ui.*;
12  import com.codename1.ui.events.*;
13  import com.codename1.ui.table.Table;
14  import com.codename1.ui.util.Resources;
15  import com.codename1.xml.Element;
16  import java.util.Vector;
17  import userclasses.util.modele.ModeleDeTableau;
18  import static userclasses.util.parser.ParserArbreXML.getLesElementsFils;
19  import static userclasses.util.parser.ParserArbreXML.getValeur;
20  import userclasses.util.parser.RequetePourArbreXML;
21  import userclasses.util.parser.RequetePourTexteBrut;
22
23  public class StateMachine extends StateMachineBase {
24
25      private RequetePourTexteBrut requeteRestConnexion;
26      private RequetePourArbreXML requeteRestCommande;
27      private Element eltCommande;
28      private String cx;
29      private Table tableau;
30      private ModeleDeTableau modeleDuTableau;
31
32      public StateMachine(String resFile) {
33          super(resFile);
34      }
35
36       @Override
37      protected void postMain(Form f) {
38      super.postMain(f); //To change body of generated methods, choose Tools | Templates.
39
40          requeteRestConnexion = new RequetePourTexteBrut();
41          requeteRestCommande = new RequetePourArbreXML();
42          //requeteRestExploitation.executer("gc/login/{id}{mdp}");
43          modeleDuTableau   = new ModeleDeTableau("Etat", "HT", "TTC","NumCom","RemiseTotale","TVA");
44          tableau = this.findTable();
45      }
46
47      @Override
48      protected void onMain_ButtonAction(Component c, ActionEvent event){
49          super.onMain_ButtonAction(c, event);
50
51          String email = this.findTextFieldEmail().getText();
52          String mdp = this.findTextFieldmdp().getText();
53
54          if (!email.equals("") && !mdp.equals("")){
55
56              requeteRestConnexion.executer("gc/client/"+email+"/"+mdp);
57              cx = requeteRestConnexion.getTexte();
58
59              System.out.println(cx);
60              //System.out.println(cx);
61              if(!cx.equals("KO")){
62
63                 requeteRestCommande.executer("gc/resumescommande/"+cx);
64                 this.eltCommande = requeteRestCommande.getRacine();
65
66                  System.out.println(eltCommande);
67                  Vector<Element>  lesElementsCommande  = getLesElementsFils(eltCommande,"dtocommande");
68                  tableau.setVisible(true);
69                  System.out.println(lesElementsCommande);
70
71                  for ( Element elementCom: lesElementsCommande){
72                      modeleDuTableau.ajouterRangee(
73                      getValeur(elementCom,"etatcom"),
74                      getValeur(elementCom,"montantht"),
75                      getValeur(elementCom,"montanttc"),
76                      getValeur(elementCom,"numcom"),
77                      getValeur(elementCom,"remisetotalnet"),
78                      getValeur(elementCom,"totaltva")
79                      );
80                  }
81                 tableau.setModel(modeleDuTableau);
82                }
83              }
84          }
85      }
```

Le fichier thème