dao.PersonneDAO

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;
import metier.modele.Personne;

/**
 *
 * @author ffonteneau
 */
public class PersonneDAO {
    public static Personne authentification(String mail, String mdp) {
        String jpql = "select␣p␣from␣Personne␣p␣where␣p.mail␣=␣:email␣and␣p.mdp␣=␣:motdepasse";
        EntityManager em = JpaUtil.obtenirEntityManager();
        Query q = em.createQuery(jpql);
        q.setParameter("email", mail);
        q.setParameter("motdepasse", mdp);
        List<Personne> res = (List<Personne>) q.getResultList();
        return res.get(0);
    }
}
```

dao.InterventionDAO

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;

import java.util.Date;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;
import metier.modele.Employe;
import metier.modele.Intervention;

/**
 *
 * @author ffonteneau
 */
public class InterventionDAO {

    public static void demandeInterv(Intervention i) {
        EntityManager em = JpaUtil.obtenirEntityManager();
        em.persist(i);
    }

    public static void cloturerIntervention(Intervention intervention){
        EntityManager em = JpaUtil.obtenirEntityManager();
        em.merge(intervention);
    }

     public static Intervention interventionCourante(Employe emp){
        String jpql = "select i from Intervention i where i.employe = :emp and i.statut = :encours";
        EntityManager em = JpaUtil.obtenirEntityManager();
        Query q = em.createQuery(jpql);
        q.setParameter("emp", emp);
        q.setParameter("encours", "EN COURS");
        List<Intervention> res = (List<Intervention>) q.getResultList();
        return res.get(0);
    }

    public static List<Intervention> historiqueJournee(){
        String jpql = "select i from Intervention i where i.statut = :encours or i.dateFin >= :date ";
        EntityManager em = JpaUtil.obtenirEntityManager();
        Query q = em.createQuery(jpql);
        Date d = new Date();
        d.setMinutes(0);
        d.setHours(0);
        q.setParameter("date", d);
        q.setParameter("encours", "EN COURS");
        List<Intervention> res = (List<Intervention>) q.getResultList();
        return res;
    }
}
```

dao.ClientDAO

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;
import metier.modele.Client;
import metier.modele.Intervention;

/**
 *
 * @author ffonteneau
 */
public class ClientDAO {

    public static void inscrire(Client c) {
        EntityManager em = JpaUtil.obtenirEntityManager();
        em.persist(c);
    }

    public static List<Intervention> historique(Client c){
        String jpql = "select␣i␣from␣Intervention␣i␣where␣i.client.id␣=␣:id";
        EntityManager em = JpaUtil.obtenirEntityManager();
        Query q = em.createQuery(jpql);
        q.setParameter("id", c.getId());
        List<Intervention> res = (List<Intervention>) q.getResultList();
        return res;
    }

}
```

dao.EmployeDAO

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;

import java.util.Date;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;
import metier.modele.Employe;
import metier.modele.Intervention;

/**
 *
 * @author ffonteneau
 */
public class EmployeDAO {

    public static void initBase() {
        Date d = new Date(1998, 9, 2);
        Employe employe = new Employe("Quentin", "Ferro", d, "homme", "0668629985", "vladimir.poutine@insa-lyon.fr", "20␣Avenue␣Albert␣Einstein,␣Lyon", true, 9, 22, "mdp");
        Employe employe2 = new Employe("Zinedine", "Zidane", d, "homme", "0606363682", "zinedine.superstar@insa-lyon.fr", "Avenue␣de␣Valescure,␣Saint-Raphaël", true, 9, 22, "mdp");
        JpaUtil.creerEntityManager();
        JpaUtil.ouvrirTransaction();
        try{
            EntityManager em = JpaUtil.obtenirEntityManager();
            em.persist(employe);
            em.persist(employe2);
            JpaUtil.validerTransaction();
        } catch(Exception ex){
            JpaUtil.annulerTransaction();
            ex.printStackTrace();  // ajouter l'envoi du mail
        }

        JpaUtil.fermerEntityManager();
    }

    public static List<Employe> employeDispo(){
        String jpql = "select␣e␣from␣Employe␣e␣where␣e.statut␣=␣1";
        EntityManager em = JpaUtil.obtenirEntityManager();
        Query q = em.createQuery(jpql);
        List<Employe> res = (List<Employe>) q.getResultList();
        return res;
    }

    public static void miseAjourEmploye(Employe emp){
        EntityManager em = JpaUtil.obtenirEntityManager();
        em.merge(emp);
    }

}
```

dao.JpaUtil

```java
package dao;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.RollbackException;

/**
 * Cette classe fournit des méthodes statiques utiles pour accéder aux
 * fonctionnalités de JPA (Entity Manager, Entity Transaction). Le nom de
 * l'unité de persistance (PERSISTENCE_UNIT_NAME) doit être conforme à la
 * configuration indiquée dans le fichier persistence.xml du projet.
 *
 * @author DASI Team
 */
public class JpaUtil {

    // *************************************************************************
    // * TODO: IMPORTANT -- Adapter le nom de l'Unité de Persistance (cf. persistence.xml) *
    // *************************************************************************
    /**
     * Nom de l'unité de persistance utilisée par la Factory de Entity Manager.
     * <br><strong>Vérifier le nom de l'unité de persistance
     * (cf.&nbsp;persistence.xml)</strong>
     */
    public static final String PERSISTENCE_UNIT_NAME = "UP";
    /**
     * Factory de Entity Manager liée à l'unité de persistance.
     * <br/><strong>Vérifier le nom de l'unité de persistance indiquée dans
     * l'attribut statique PERSISTENCE_UNIT_NAME
     * (cf.&nbsp;persistence.xml)</strong>
     */
    private static EntityManagerFactory entityManagerFactory = null;
    /**
     * Gère les instances courantes de Entity Manager liées aux Threads.
     * L'utilisation de ThreadLocal garantie une unique instance courante par
     * Thread.
     */
    private static final ThreadLocal<EntityManager> threadLocalEntityManager = new ThreadLocal<
        EntityManager>() {

        @Override
        protected EntityManager initialValue() {
            return null;
        }
    };

    // Méthode pour avoir des messages de Log dans le bon ordre (pause)
    private static void pause(long milliseconds) {
        try {
            Thread.sleep(milliseconds);
        } catch (InterruptedException ex) {
            ex.hashCode();
        }
    }

    // Méthode pour avoir des messages de Log dans le bon ordre (log)
    private static void log(String message) {
        System.out.flush();
        pause(5);
        System.err.println("[JpaUtil:Log]␣" + message);
        System.err.flush();
        pause(5);
    }

    /**
     * Initialise la Factory de Entity Manager.
     * <br><strong>À utiliser uniquement au début de la méthode main() [projet
     * Java Application] ou dans la méthode init() de la Servlet Contrôleur
     * (ActionServlet) [projet Web Application].</strong>
     */
    public static synchronized void init() {
        log("Initialisation␣de␣la␣factory␣de␣contexte␣de␣persistance");
        if (entityManagerFactory != null) {
            entityManagerFactory.close();
        }
        entityManagerFactory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
    }

    /**
     * Libère la Factory de Entity Manager.
     * <br><strong>À utiliser uniquement à la fin de la méthode main() [projet
     * Java Application] ou dans la méthode destroy() de la Servlet Contrôleur
     * (ActionServlet) [projet Web Application].</strong>
     */
    public static synchronized void destroy() {
        log("Libération␣de␣la␣factory␣de␣contexte␣de␣persistance");
        if (entityManagerFactory != null) {
            entityManagerFactory.close();
            entityManagerFactory = null;
        }
    }

    /**
     * Créée l'instance courante de Entity Manager (liée à ce Thread).
     * <br><strong>À utiliser uniquement au niveau Service.</strong>
     */
    public static void creerEntityManager() {
        log("Création␣du␣contexte␣de␣persistance");
        threadLocalEntityManager.set(entityManagerFactory.createEntityManager());
    }

    /**
     * Ferme l'instance courante de Entity Manager (liée à ce Thread).
     * <br><strong>À utiliser uniquement au niveau Service.</strong>
     */
    public static void fermerEntityManager() {
        log("Fermeture␣du␣contexte␣de␣persistance");
        EntityManager em = threadLocalEntityManager.get();
        em.close();
        threadLocalEntityManager.set(null);
    }

    /**
     * Démarre une transaction sur l'instance courante de Entity Manager.
     * <br><strong>À utiliser uniquement au niveau Service.</strong>
     */
    public static void ouvrirTransaction() {
        log("Ouverture␣de␣la␣transaction␣(begin)");
        try {
            EntityManager em = threadLocalEntityManager.get();
            em.getTransaction().begin();
        } catch (Exception ex) {
            log("Erreur␣lors␣de␣l'ouverture␣de␣la␣transaction");
            throw ex;
        }
    }

    /**
     * Valide la transaction courante sur l'instance courante de Entity Manager.
     * <br><strong>À utiliser uniquement au niveau Service.</strong>
     *
     * @exception RollbackException lorsque le <em>commit</em> n'a pas réussi.
     */
    public static void validerTransaction() throws RollbackException {
        log("Validation␣de␣la␣transaction␣(commit)");
        try {
            EntityManager em = threadLocalEntityManager.get();
            em.getTransaction().commit();
        } catch (Exception ex) {
            log("Erreur␣lors␣de␣la␣validation␣(commit)␣de␣la␣transaction");
            throw ex;
        }
    }

    /**
     * Annule la transaction courante sur l'instance courante de Entity Manager.
     * Si la transaction courante n'est pas démarrée, cette méthode n'effectue
     * aucune opération.
     * <br><strong>À utiliser uniquement au niveau Service.</strong>
     */
    public static void annulerTransaction() {
        try {
            log("Annulation␣de␣la␣transaction␣(rollback)");

            EntityManager em = threadLocalEntityManager.get();
            if (em.getTransaction().isActive()) {
                log("Annulation␣effective␣de␣la␣transaction␣(rollback␣d'une␣transaction␣active)");
                em.getTransaction().rollback();
            }

        } catch (Exception ex) {
            log("Erreur␣lors␣de␣l'annulation␣(rollback)␣de␣la␣transaction");
            throw ex;
        }
    }

    /**
     * Retourne l'instance courante de Entity Manager.
     * <br><strong>À utiliser uniquement au niveau DAO.</strong>
     *
     * @return instance de Entity Manager
     */
    protected static EntityManager obtenirEntityManager() {
        log("Obtention␣du␣contexte␣de␣persistance");
        return threadLocalEntityManager.get();
    }
}
```

util.Saisie

```java
package util;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.List;

/**
 *
 * @author DASI Team
 */
public class Saisie {

    public static String lireChaine(String invite) {
        String chaineLue = null;
        System.out.print(invite);
        try {
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            chaineLue = br.readLine();
        } catch (IOException ex) {
            ex.printStackTrace(System.err);
        }
        return chaineLue;

    }

    public static Integer lireInteger(String invite) {
        Integer valeurLue = null;
        while (valeurLue == null) {
            try {
                valeurLue = Integer.parseInt(lireChaine(invite));
            } catch (NumberFormatException ex) {
                System.out.println("/!\\ Erreur de saisie - Nombre entier attendu /!\\");
            }
        }
        return valeurLue;
    }

    public static Integer lireInteger(String invite, List<Integer> valeursPossibles) {
        Integer valeurLue = null;
        while (valeurLue == null) {
            try {
                valeurLue = Integer.parseInt(lireChaine(invite));
            } catch (NumberFormatException ex) {
                System.out.println("/!\\ Erreur de saisie - Nombre entier attendu /!\\");
            }
            if (!(valeursPossibles.contains(valeurLue))) {
                System.out.println("/!\\ Erreur de saisie - Valeur non-autorisée /!\\");
                valeurLue = null;
            }
        }
        return valeurLue;
    }

    public static void pause() {
        lireChaine("--PAUSE--");
    }

    public static void main(String[] args) {

        System.out.println("Bonjour !");

        String nom = Saisie.lireChaine("Entrez votre nom : ");
        System.out.println("Bonjour, " + nom + " !");

        Integer age = Saisie.lireInteger("Entrez votre âge : ");
        System.out.println("Vous avez " + age + " ans.");

        Integer annee = Saisie.lireInteger("Entrez votre année au Département IF (3,4,5): ", Arrays.asList(3,4,5));
        System.out.println("Vous êtes en " + annee + "IF.");

        Saisie.pause();

        System.out.println("Au revoir !");
    }
}
```

util.Message

```java
package util;

import java.io.PrintStream;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.text.SimpleDateFormat;
import java.util.Date;

/**
 *
 * @author DASI Team
 */
public class Message {

    private final static PrintStream OUT = System.out;
    private final static SimpleDateFormat TIMESTAMP_FORMAT = new SimpleDateFormat("yyyy-MM-dd~HH:mm:ss");
    private final static SimpleDateFormat HORODATE_FORMAT = new SimpleDateFormat("dd/MM/yyyy␣à␣HH:mm:ss");

    private static void debut() {
        Date maintenant = new Date();
        OUT.println();
        OUT.println();
        OUT.println("---<([␣MESSAGE␣@␣" + TIMESTAMP_FORMAT.format(maintenant) + "␣])>---");
        OUT.println();
    }

    private static void fin() {
        OUT.println();
        OUT.println("---<([␣FIN␣DU␣MESSAGE␣])>---");
        OUT.println();
        OUT.println();
    }

    public static void envoyerMail(String mailExpediteur, String mailDestinataire, String objet, String corps) {

        Date maintenant = new Date();
        Message.debut();
        OUT.println("~~~␣E-mail␣envoyé␣le␣" + HORODATE_FORMAT.format(maintenant) + "␣~~~");
        OUT.println("De␣:␣" + mailExpediteur);
        OUT.println("À␣␣:␣" + mailDestinataire);
        OUT.println("Obj:␣" + objet);
        OUT.println();
        OUT.println(corps);
        Message.fin();
    }

    public static void envoyerNotification(String telephoneDestinataire, String message) {

        Date maintenant = new Date();
        Message.debut();
        OUT.println("~~~␣Notification␣envoyée␣le␣" + HORODATE_FORMAT.format(maintenant) + "␣~~~");
        OUT.println("À␣␣:␣" + telephoneDestinataire);
        OUT.println();
        OUT.println(message);
        Message.fin();
    }

    public static void main(String[] args) {

        //DebugLogger.log("Début des Tests...");

        StringWriter corps = new StringWriter();
        PrintWriter mailWriter = new PrintWriter(corps);

        mailWriter.println("Bonjour,");
        mailWriter.println();
        mailWriter.println("␣␣Ceci␣est␣un␣mail␣destiné␣à␣tester␣l'envoi␣simulé␣par␣affichage␣sur␣la␣console.");
        mailWriter.println();
        mailWriter.println("␣␣Cordialement,");
        mailWriter.println();
        mailWriter.println("␣␣␣␣Yann␣Gripay");

```

```java
        Message.envoyerMail(
                "yann.gripay@insa-lyon.fr",
                "etudiants.3IF@insa-lyon.fr",
                "[DASI]␣Test␣d'envoi␣de␣e-mail",
                corps.toString()
            );


        StringWriter message = new StringWriter();
        PrintWriter notificationWriter = new PrintWriter(message);

        notificationWriter.println("Ceci␣est␣une␣notification␣pour␣prévenir␣de␣2␣choses:");
        notificationWriter.println("1)␣NE␣PAS␣oublier␣le␣poly");
        notificationWriter.println("2)␣TESTER␣au␣fur␣et␣à␣mesure␣du␣développement");

        Message.envoyerNotification(
                "0988776655",
                message.toString()
            );

        //DebugLogger.log("Fin des Tests...");

    }
}
```

util.DebugLogger

```java
package util;

/**
 *
 * @author DASI Team
 */
public class DebugLogger {

    // Méthode pour avoir des messages de Log dans le bon ordre (pause)
    public static void pause(long milliseconds) {
        try {
            Thread.sleep(milliseconds);
        } catch (InterruptedException ex) {
            ex.hashCode();
        }
    }

    // Méthode pour avoir des messages de Log dans le bon ordre (log)
    public static void log(String message) {
        System.out.flush();
        pause(5);
        System.err.println("[DebugLogger]␣" + message);
        System.err.flush();
        pause(5);
    }

    // Méthode pour avoir des messages de Log dans le bon ordre (log avec exception)
    public static void log(String message, Exception ex) {
        System.out.flush();
        pause(5);
        System.err.println("[DebugLogger]␣" + message);
        System.err.println("[**EXCEPTION**]␣" + ex.getMessage());
        System.err.print("[**EXCEPTION**]␣>>>␣");
        ex.printStackTrace(System.err);
        System.err.flush();
        pause(5);
    }

    public static void main(String[] args) {

        DebugLogger.log("**␣Début␣du␣Test␣**");

        DebugLogger.log("Message␣de␣DEBUG␣pour␣tester...");

        Integer a = 0;
        Integer b = null;

        if (a < 0) {
            DebugLogger.log("ERREUR␣sur␣la␣valeur␣de␣A");
        }

        try {
            if (b >  0) {
                DebugLogger.log("Test␣OK␣pour␣la␣valeur␣de␣B");
            }
        } catch (Exception ex) {
            DebugLogger.log("Problème␣avec␣B", ex);
        }

        DebugLogger.log("**␣Fin␣du␣Test␣**");
    }
}
```

util.GeoTest

```java
package util;

import com.google.maps.DirectionsApi;
import com.google.maps.DirectionsApiRequest;
import com.google.maps.GeoApiContext;
import com.google.maps.GeocodingApi;
import com.google.maps.model.DirectionsResult;
import com.google.maps.model.DirectionsRoute;
import com.google.maps.model.GeocodingResult;
import com.google.maps.model.LatLng;
import com.google.maps.model.TravelMode;

/**
 *
 * @author DASI Team
 */
/* DÉPENDANCES Maven:
<dependency>
    <groupId>com.google.maps</groupId>
    <artifactId>google-maps-services</artifactId>
    <version>0.2.11</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-nop</artifactId>
    <version>1.7.26</version>
</dependency>
*/
public class GeoTest {

    final static String MA_CLE_GOOGLE_API = "AIzaSyAhf3JleYpal9S-xouJYH8lf7Dvz5Y2Nko";

    final static GeoApiContext MON_CONTEXTE_GEOAPI = new GeoApiContext.Builder().apiKey(MA_CLE_GOOGLE_API).build();

    public static LatLng getLatLng(String adresse) {
        try {
            GeocodingResult[] results = GeocodingApi.geocode(MON_CONTEXTE_GEOAPI, adresse).await();

            return results[0].geometry.location;

        } catch (Exception ex) {
            return null;
        }
    }

    public static double toRad(double angleInDegree) {
        return angleInDegree * Math.PI / 180.0;
    }

    public static double getFlightDistanceInKm(LatLng origin, LatLng destination) {

        // From: http://www.movable-type.co.uk/scripts/latlong.html
        double R = 6371.0; // Average radius of Earth (km)
        double dLat = toRad(destination.lat - origin.lat);
        double dLon = toRad(destination.lng - origin.lng);
        double lat1 = toRad(origin.lat);
        double lat2 = toRad(destination.lat);

        double a = Math.sin(dLat / 2.0) * Math.sin(dLat / 2.0)
                + Math.sin(dLon / 2.0) * Math.sin(dLon / 2.0) * Math.cos(lat1) * Math.cos(lat2);
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1.0 - a));
        double d = R * c;

        return Math.round(d * 1000.0) / 1000.0;
    }

    public static Double getTripDurationByBicycleInMinute(LatLng origin, LatLng destination, LatLng... steps) {
        return getTripDurationOrDistance(TravelMode.BICYCLING, true, origin, destination, steps);
    }

    public static Double getTripDistanceByCarInKm(LatLng origin, LatLng destination, LatLng... steps) {
        return getTripDurationOrDistance(TravelMode.DRIVING, false, origin, destination, steps);
    }

    public static Double getTripDurationOrDistance(TravelMode mode, boolean duration, LatLng origin,
        LatLng destination, LatLng... steps) {

        DirectionsApiRequest request = DirectionsApi.getDirections(MON_CONTEXTE_GEOAPI, origin.toString()
            , destination.toString());
        request.mode(mode);
        request.region("fr");

        if (steps.length > 0) {

            String[] stringSteps = new String[steps.length];
            for (int i = 0; i < steps.length; i++) {
                stringSteps[i] = steps[i].toString();
            }

            request.waypoints(stringSteps);
        }

        double cumulDistance = 0.0;
        double cumulDuration = 0.0;

        try {
            DirectionsResult result = request.await();
            DirectionsRoute[] directions = result.routes;

            for (int legIndex = 0; legIndex < directions[0].legs.length; legIndex++) {

                cumulDistance += directions[0].legs[legIndex].distance.inMeters / 1000.0;
                cumulDuration += Math.ceil(directions[0].legs[legIndex].duration.inSeconds / 60.0);
            }

        } catch (Exception ex) {
            return null;
        }

        if (duration) {
            return cumulDuration;
        } else {
            return cumulDistance;
        }
    }

    public static void main(String[] args) {

        if (MA_CLE_GOOGLE_API.equals("XXXXXXXX-Moodle-Clé")) {
            for (int i=0; i<100; i++) {
                System.err.println("[ERREUR]␣VOUS␣AVEZ␣OUBLIÉ␣DE␣CHANGER␣LA␣CLÉ␣DE␣L'API␣!!!!!");
            }
            System.exit(-1);
        }

        String adresse1 = "7␣Avenue␣Jean␣Capelle␣Ouest,␣Villeurbanne";
        LatLng coords1 = getLatLng(adresse1);
        System.out.println("Lat/Lng␣de␣Adresse␣#1:␣" + coords1);

        String adresse2 = "37␣Avenue␣Jean␣Capelle␣Est,␣Villeurbanne";
        LatLng coords2 = getLatLng(adresse2);
        System.out.println("Lat/Lng␣de␣Adresse␣#2:␣" + coords2);

        String adresse3 = "61␣Avenue␣Roger␣Salengro,␣Villeurbanne";
        LatLng coords3 = getLatLng(adresse3);
        System.out.println("Lat/Lng␣de␣Adresse␣#3:␣" + coords3);

        // Coordonnées directes: Rond-Point du Totem, Cours Tolstoï, Villeurbanne
        LatLng coords4 = new LatLng(45.763781,4.8735128);
        System.out.println("Lat/Lng␣de␣Coords␣#4:␣(␣" + coords4.lat + ";␣" + coords4.lng + "␣)");


        Double duree = getTripDurationByBicycleInMinute(coords1, coords3);
        System.out.println("Durée␣de␣Trajet␣à␣Vélo␣de␣Adresse␣#1␣à␣Adresse␣#3␣(trajet␣direct):␣" + duree
            + "␣min");

        Double distance = getTripDistanceByCarInKm(coords1, coords3, coords2);
        System.out.println("Distance␣en␣Voiture␣de␣Adresse␣#1␣à␣Adresse␣#3␣en␣passant␣par␣Adresse␣#2␣(
            distance␣par␣la␣route):␣" + distance + "␣km");

        Double distanceVolDOiseau = getFlightDistanceInKm(coords1, coords3);
        System.out.println("Distance␣à␣Vol␣d'Oiseau␣de␣Adresse␣#1␣à␣Adresse␣#3␣(distance␣géographique):␣"
            + distanceVolDOiseau + "␣km");

        Double autreDistanceVolDOiseau = getFlightDistanceInKm(coords1, coords4);
        System.out.println("Distance␣à␣Vol␣d'Oiseau␣de␣Adresse␣#1␣à␣Coords␣#4␣(distance␣géographique):␣"
           + autreDistanceVolDOiseau + "␣km");
    }
}
```

metier.modele.Personne

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier.modele;

import java.io.Serializable;
import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Inheritance;
import javax.persistence.InheritanceType;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 *
 * @author ffonteneau
 */
@Entity
@Inheritance(strategy = InheritanceType.JOINED)
public abstract class  Personne implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    protected Integer id;
    protected String nom;
    protected String prenom;
    @Temporal(TemporalType.DATE) protected Date dateNaissance;
    protected String civilite;
    protected String numTel;
    protected String mail;
    protected String adresse;
    protected String mdp;

    public Personne(){}

    public Personne(String nom, String prenom, Date date, String civilite,
            String numTel, String mail, String adresse, String mdp) {
        this.nom = nom;
        this.prenom = prenom;
        this.dateNaissance = date;
        this.civilite = civilite;
        this.numTel = numTel;
        this.mail = mail;
        this.adresse = adresse;
        this.mdp = mdp;
    }

    public String getNom() {
        return nom;
    }

    public String getPrenom() {
        return prenom;
    }

    public Date getDateNaissance() {
        return dateNaissance;
    }

    public String getCivilite() {
        return civilite;
    }

    public String getNumTel() {
        return numTel;
    }

    public String getMail() {
        return mail;
    }

    public String getAdresse() {
        return adresse;
    }

    public Integer getId() {
        return id;
    }

    public String getMdp() {
        return mdp;
    }

    public void setDateNaissance(Date dateNaissance) {
        this.dateNaissance = dateNaissance;
    }

    public void setCivilite(String civilite) {
        this.civilite = civilite;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }
}
```

metier.modele.Intervention

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier.modele;

import java.io.Serializable;
import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Inheritance;
import javax.persistence.InheritanceType;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 *
 * @author ffonteneau
 */

@Entity
@Inheritance(strategy = InheritanceType.JOINED)
public abstract class Intervention implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    protected Integer id;
    protected Client client;
    protected Employe employe;
    @Temporal(TemporalType.DATE) protected Date dateInter;
    @Temporal(TemporalType.DATE) protected Date dateFin;
    protected String description;
    protected String statut;
    protected String commentaire;
    public Intervention(Client client, Date dateInter,
            String description, String statut){
        this.client = client;
        this.employe = employe;
        this.dateInter = dateInter;
        this.description = description;
        this.statut = statut;
    }

    public Intervention() {
    }

    public Date getDateFin() {
        return dateFin;
    }

    public void setCommentaire(String commentaire) {
        this.commentaire = commentaire;
    }

    public String getCommentaire() {
        return commentaire;
    }

    public String getStatut() {
        return statut;
    }

    public void setDateFin(Date dateFin) {
        this.dateFin = dateFin;
    }

    public void setStatut(String statut) {
        this.statut = statut;
    }


    public Client getClient() {
        return client;
    }

    public Employe getEmploye() {
        return employe;
    }

    public Date getDateInter() {
        return dateInter;
    }

    public String getDescription() {
        return description;
    }

    public Integer getId() {
        return id;
    }

    public void setClient(Client client) {
        this.client = client;
    }

    public void setEmploye(Employe employe) {
        this.employe = employe;
    }

    public void setDateInter(Date dateNaissance) {
        this.dateInter = dateNaissance;
    }

    public void setDescription(String description) {
        this.description = description;
    }

}
```

metier.modele.Employe

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier.modele;

import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.Transient;


/**
 *
 * @author ffonteneau
 */

@Entity
//public class Employe  extends Personne {
public class Employe extends Personne {
    protected Integer numEmploye;
    protected boolean statut;
    protected Integer heureDebut;
    protected Integer heureFin;

    public Employe (){}

    public Employe(String nom, String prenom, Date date, String civilite,
            String numTel, String mail, String adresse,
            boolean statut, Integer heureDebut, Integer heureFin, String mdp){
        super(nom, prenom, date, civilite, numTel, mail, adresse, mdp);
        this.statut = statut;
        this.heureDebut = heureDebut;
        this.heureFin = heureFin;
        this.numEmploye = id;
    }

    public Integer getNumEmploye() {
        return numEmploye;
    }

    public boolean isStatut() {
        return statut;
    }

    public Integer getHeureDebut() {
        return heureDebut;
    }

    public Integer getHeureFin() {
        return heureFin;
    }

    public void setNumEmploye(Integer numEmploye) {
        this.numEmploye = numEmploye;
    }

    public void setStatut(boolean statut) {
        this.statut = statut;
    }

    public void setHeureDebut(Integer heureDebut) {
        this.heureDebut = heureDebut;
    }

    public void setHeureFin(Integer heureFin) {
        this.heureFin = heureFin;
    }


}
```

metier.modele.InterventionAnimale

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier.modele;

import java.util.Date;
import javax.persistence.Entity;

/**
 *
 * @author ffonteneau
 */
@Entity
public class InterventionAnimale extends Intervention {
    protected String typeAnimal;

    public InterventionAnimale(Client client, Date dateInter,
              String description, String typeAnimal, String statut){
        super(client, dateInter, description, statut);
        this.typeAnimal = typeAnimal;
    }

    public InterventionAnimale(){}

    public String getTypeAnimal() {
        return typeAnimal;
    }

    public void setTypeAnimal(String typeAnimal) {
        this.typeAnimal = typeAnimal;
    }
}
```

metier.modele.InterventionIncident

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier.modele;

import java.util.Date;
import javax.persistence.Entity;

/**
 *
 * @author ffonteneau
 */
@Entity
public class InterventionIncident extends Intervention {

    public InterventionIncident(Client client, Date dateInter,
             String description, String statut){
        super(client, dateInter, description, statut);
    }

    public InterventionIncident() {
    }


}
```

metier.modele.Client

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier.modele;

import java.util.Date;
import javax.persistence.Entity;

/**
 *
 * @author ffonteneau
 */

@Entity
public class Client extends Personne {

    public Client (){}

    public Client(String nom, String prenom, Date date, String civilite, String numTel, String mail,
        String adresse, String mdp){
        super(nom, prenom, date, civilite, numTel, mail, adresse, mdp);
    }
}
```

metier.modele.InterventionLivraison

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier.modele;

import java.util.Date;
import javax.persistence.Entity;

/**
 *
 * @author ffonteneau
 */
@Entity
public class InterventionLivraison extends Intervention {
    protected String objet;
    protected String entreprise;

    public InterventionLivraison(String objet, String entreprise, Client client,
             Date dateInter, String description, String statut){
        super(client, dateInter, description, statut);
        this.objet = objet;
        this.entreprise = entreprise;
    }

    public InterventionLivraison() {
    }

    public String getObjet() {
        return objet;
    }

    public String getEntreprise() {
        return entreprise;
    }

    public void setObjet(String objet) {
        this.objet = objet;
    }

    public void setEntreprise(String entreprise) {
        this.entreprise = entreprise;
    }
}
```

metier.service.Service

```java
package metier.service;

import com.google.maps.model.LatLng;
import dao.ClientDAO;
import dao.EmployeDAO;
import dao.InterventionDAO;
import dao.JpaUtil;
import dao.PersonneDAO;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.util.Date;
import java.util.List;
import metier.modele.Client;
import metier.modele.Employe;
import metier.modele.Intervention;
import metier.modele.Personne;
import util.DebugLogger;
import static util.GeoTest.getLatLng;
import static util.GeoTest.getTripDurationByBicycleInMinute;
import util.Message;

/**
 *
 * @author ffonteneau
 */
public class Service {


    public static boolean inscription(Client c) {
        //System.out.println("[Service:Log] inscription de "+c.getNom());
        boolean inscrit = true;
        JpaUtil.creerEntityManager();
        JpaUtil.ouvrirTransaction();
        try{
            ClientDAO.inscrire(c);
            JpaUtil.validerTransaction();
        } catch(Exception e){
            inscrit = false;
            DebugLogger.log("[Service:Log] echec d'inscription de "+c.getNom(), e);
            JpaUtil.annulerTransaction();
        }
        if (inscrit){
                        //envoie du mail
            StringWriter corps = new StringWriter();
            PrintWriter mailWriter = new PrintWriter(corps);

            mailWriter.println("Bonjour "+c.getPrenom()+",");
            mailWriter.println();
            mailWriter.println("  Ceci est un mail vous confirmant votre inscription à PROACT'IF !");
            mailWriter.println();
            mailWriter.println("  Cordialement,");
            mailWriter.println();
            mailWriter.println("    L'équipe PROACT'IF");

            Message.envoyerMail(
                "pro@actif.fr",
                c.getMail(),
                "[PROACT'IF] Confirmation inscription",
                corps.toString()
            );
            //System.out.println("[Service:Log] inscription de "+c.getNom()+" terminée");
        }
        JpaUtil.fermerEntityManager();
        return inscrit;
    }

    //méthode servant à rechercher un employe pour une inscription
    //la recherche s'effectue sur
    private static Employe rechercherEmploye(Intervention i){
        List<Employe> listeEmploye;
        Employe emp = null;
        JpaUtil.creerEntityManager();
        listeEmploye = EmployeDAO.employeDispo();
        if(listeEmploye != null){
            //dans tous les employes disponibles rechrcher le plus proche
            emp = listeEmploye.get(0);
            for(Employe e : listeEmploye){
                LatLng coordsClient = getLatLng(i.getClient().getAdresse());
                LatLng coordsCurrent = getLatLng(emp.getAdresse());
                LatLng coordsComp = getLatLng(e.getAdresse());
                if(getTripDurationByBicycleInMinute(coordsClient, coordsComp) <
                    getTripDurationByBicycleInMinute(coordsClient, coordsCurrent)) {
                    emp = e;
                }
            }
        }
        JpaUtil.fermerEntityManager();
        return emp;
    }

    //prend une intervention sans employe mais avec un client,
    //trouve l'employe correspondant
    public static boolean demandeInter(Intervention inter) {
        //System.out.println("[Service:Log] "+inter.getClient().getNom()+" demande intervention");
        Employe emp = rechercherEmploye(inter);
        boolean interventionAcceptee = true;
        if (emp != null){
            inter.setEmploye(emp);
            inter.setStatut("EN␣COURS");
            emp.setStatut(false);

            JpaUtil.creerEntityManager();
            JpaUtil.ouvrirTransaction();

            try{
                InterventionDAO.demandeInterv(inter);
                EmployeDAO.miseAjourEmploye(emp);
                JpaUtil.validerTransaction();

                //notification de l'intervention de l'employe par message
                StringWriter message = new StringWriter();
                PrintWriter notificationWriter = new PrintWriter(message);
                notificationWriter.println("Bonjour,␣"+emp.getPrenom());
                notificationWriter.println("Vous␣avez␣une␣nouvelle␣intervention␣!");
                String typeInter = inter.getClass().getName();
                typeInter = typeInter.substring(26);
                notificationWriter.println("C'est␣une␣intervention␣de␣type␣:␣"+typeInter);
                notificationWriter.println("Adresse␣:␣"+inter.getClient().getAdresse());
                notificationWriter.println("Descriptif␣:␣"+inter.getDescription());
                Message.envoyerNotification(
                    emp.getNumTel(),
                    message.toString()
                );
            } catch(Exception e){
                JpaUtil.annulerTransaction();
                interventionAcceptee = false;
                DebugLogger.log("[Service:Log]␣erreur␣d'ajout␣d'intervention␣:␣",e);

            }
            JpaUtil.fermerEntityManager();

        } else {
            interventionAcceptee = false;
            DebugLogger.log("[Service:Log]␣Intervention␣refusée,␣aucun␣employe␣disponible");
        }
        return interventionAcceptee;
    }

    //Envoie pointernull si Personne non trouvée
    public static Personne authentification(String mail, String mdp) {
        Personne personne = null;
        JpaUtil.creerEntityManager();
        try {
            personne = PersonneDAO.authentification(mail, mdp);
            //System.out.println("[Service:Log] "+personne.getNom()+" connecté");
        } catch(Exception e) {
            DebugLogger.log("[Service:Log]␣echec␣de␣connexion␣de␣"+mail);
            //authentification non complete
        }
        JpaUtil.fermerEntityManager();
        return personne;
    }


    //récupère l'historique des intervention d'un client, renvoie une
    //une liste vide ou pointeur null si aucune intervention
    public static List<Intervention> historiqueClient(Client c) {
        //System.out.println("[Service:Log] demande historique de "+c.getNom());
        List<Intervention> historique = null;
        JpaUtil.creerEntityManager();
        try{
            historique = ClientDAO.historique(c);
        }catch(Exception e){
            DebugLogger.log("[Service:Log]␣Erreur␣en␣cherchant␣l'historique␣du␣client␣:␣",e);
        }
        JpaUtil.fermerEntityManager();
        return historique;
    }

    //Renvoie l'intervention que l'employe a à effectuer,
    //Renvoie null si pas d'intervention à faire
    public static Intervention interventionActuelle(Employe emp){
        Intervention intervention = null;
        JpaUtil.creerEntityManager();

        //récupération de l'intervention
        try {
            intervention = InterventionDAO.interventionCourante(emp);
            //System.out.println("[Service:Log] Cloture de l'intervention");
        } catch(Exception e) {
            DebugLogger.log("[Service:Log]␣Erreur␣en␣trouvant␣l'intervention␣de␣l'employe␣:␣",e);
        }
        JpaUtil.fermerEntityManager();
        return intervention;
    }

    //cloture l'intervention actuelle d'un employe
    public static boolean cloturerIntervention(Employe emp, String commentaire, String statut){
        boolean cloturee = true;
        Intervention intervention = interventionActuelle(emp);

        //mise à jour de l'intervention et du statut de l'employe
        if(intervention != null){
            intervention.setStatut(statut);
            intervention.setCommentaire(commentaire);
            intervention.setDateFin(new Date());
            emp.setStatut(true);

            JpaUtil.creerEntityManager();
            JpaUtil.ouvrirTransaction();
            try{
                InterventionDAO.cloturerIntervention(intervention);
                EmployeDAO.miseAjourEmploye(emp);
                //System.out.println("[Service:Log] Intervention cloturée");
                JpaUtil.validerTransaction();

                //notification du client
                StringWriter message = new StringWriter();
                PrintWriter notificationWriter = new PrintWriter(message);
                notificationWriter.println("Bonjour,␣"+intervention.getClient().getPrenom());
                notificationWriter.println("Votre␣intervention␣a␣bien␣été␣traitée,");
                notificationWriter.println("Commentaire␣de␣l'employé␣:␣"+intervention.getCommentaire());
                Message.envoyerNotification(
                    intervention.getClient().getNumTel(),
                    message.toString()
                );
            }catch(Exception e){
                cloturee = false;
                JpaUtil.annulerTransaction();
                DebugLogger.log("[Service:Log]␣Erreur␣de␣cloture␣d'intervention",e);
            }
            JpaUtil.fermerEntityManager();
        }else{
            cloturee = false;
            DebugLogger.log("[Service:Log]␣Employe␣n'a␣pas␣d'intervention␣courante");
        }
        return cloturee;
    }

    //envoie une liste de toutes les interventions de la journée
    public static List<Intervention> interventionsJournee(){
        List<Intervention> historique = null;
        JpaUtil.creerEntityManager();
        try{
            historique = InterventionDAO.historiqueJournee();
        }catch(Exception e){
            DebugLogger.log("[Service:Log]␣Erreur␣en␣cherchant␣l'historique␣du␣client␣:␣",e);
        }
        JpaUtil.fermerEntityManager();
        return historique;
    }
}
```

vue.Main

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package vue;

import com.google.maps.model.LatLng;
import util.GeoTest;
import dao.EmployeDAO;
import dao.JpaUtil;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.util.Date;
import java.util.List;
import javax.persistence.EntityManager;
import metier.modele.*;
import metier.service.*;
import util.DebugLogger;
import static util.GeoTest.getFlightDistanceInKm;
import static util.GeoTest.getLatLng;
import static util.GeoTest.getTripDistanceByCarInKm;
import static util.GeoTest.getTripDurationByBicycleInMinute;
import util.Message;

/**
 *
 * @author Quentin Ferro
 */
public class Main {

    public static void main(String[] args) {

        //exemple d'utilisation des méthodes de la couche service
        JpaUtil.init();

        //initialisation des employes dans la base
        EmployeDAO.initBase();

        //ajout de quelques clients dans la base
        Date d = new Date(1998, 9, 2);
        Client salut = new Client("Ferro", "BG", d, "CHOSE", "0668629985", "mao.zedong@insa-lyon.fr", "221␣Boulevard␣Adrian,␣Saint-Raphael", "mdp");
        Client salut2 = new Client("Lechat", "Felix", d, "CHOSE", "0609837263", "lenine@insa-lyon.fr", "Rue␣de␣la␣République,␣Lyon", "mdp");
        Service.inscription(salut);
        Service.inscription(salut2);


        //authentification client
        Service.authentification("mao.zedong@insa-lyon.fr", "mdp");
        Personne p = Service.authentification("mao.zedong@insa-lyon.fr", "hmmm");
        Service.authentification("bah", "mdp");

        //authentification employe
        Employe emp = (Employe) Service.authentification( "vladimir.poutine@insa-lyon.fr",  "mdp");

        //ajout d'intervention à la base
        InterventionAnimale interMilan = new InterventionAnimale(salut, new Date(), "blabla", "Chat" ,"test");
        InterventionAnimale inter2 = new InterventionAnimale(salut2, new Date(), "miaou", "Chat" ,"test");
        System.out.println("====="+interMilan.getClass().getName());
        Service.demandeInter(interMilan);
        Service.demandeInter(inter2);

        //liste de l'historique du client
        List<Intervention> historique = Service.historiqueClient(salut);
        for(Intervention i : historique){
            System.out.println("␣␣␣"+i.getDescription());
        }

        //cloture d'une intervention
        Service.cloturerIntervention(emp, "c'était␣très␣bien", "Terminéee");

        //historique des intervention de la journées
        List<Intervention> historique2 = Service.interventionsJournee();
        for(Intervention i : historique2){
            System.out.println("␣␣␣"+i.getDescription());
        }


        JpaUtil.destroy();
    }
}
```