Projet Algorithmique et programmation java avec Mme AUBONNET Tatiana

2009

STAG

Analyste Concepteur des Systèmes d'Information et de Décision

1 SOMMAIRE

2 INTRODUCTION

C’est un projet qui fait parti dans notre formation au Conservatoire National des Arts et Métier.

Ce projet a pour objet la réalisation d’une application de gestion des rendez-vous, des contacts, des taches ainsi que des alertes. Cette application sera développée en java. Il a été réalisé par M. LANDAIS Rémi et M. CHEMALY Alain, nous avons choisi de la baptisé STAG.

Responsable de cette discipline Mme AUBONNET Tatiana, qui nous a suivis tout le long de la réalisation du projet.

2.1 Description globale :

Le projet sera composé de cinq modules :

- Un module Calendrier
- Un module de Gestion des contacts
- Un module de Gestion des RDV
- Un module de Gestion des taches
- Un module « pense-bête » alertes

2.2 Description détaillée :

Module calendrier : Ce module va servir de calendrier à l'utilisateur. On pourra y voir la date du jour ainsi que l'heure.

Module Gestion des contacts : Ce module se rapproche de la gestion des contacts d'un téléphone portable. On y retrouvera le nom, prénom de la personne, son numéro de fixe, de portable ainsi que son numéro professionnel; on pourra également faire apparaitre une photo du contact.

Module gestion des RDV: Ce module va permettre à l'utilisateur de notifier tous ses rendez-vous.

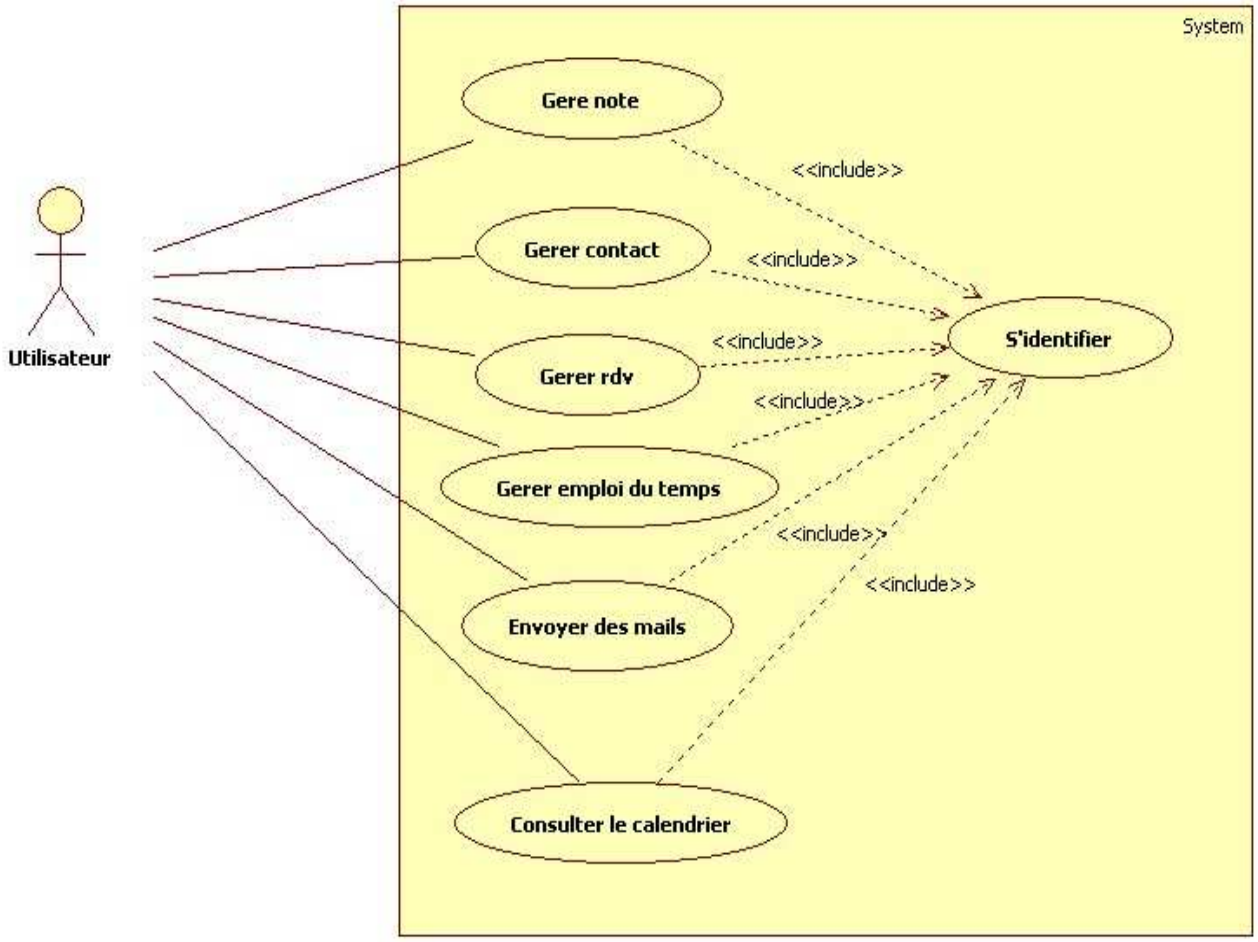
Module gestion des taches: Ce module peut s'apparenter à un emploi du temps, il permet à l'utilisateur de programmer sa journée, sa semaine.

Module « pense-bête »: Grâce à ce module l'utilisateur va pouvoir créer des alertes qu'il pourra programmer pour un certain jour et une certaine heure afin de ne pas oublier certaines taches importantes à effectuer.

3 CONCEPTION DU PROJET

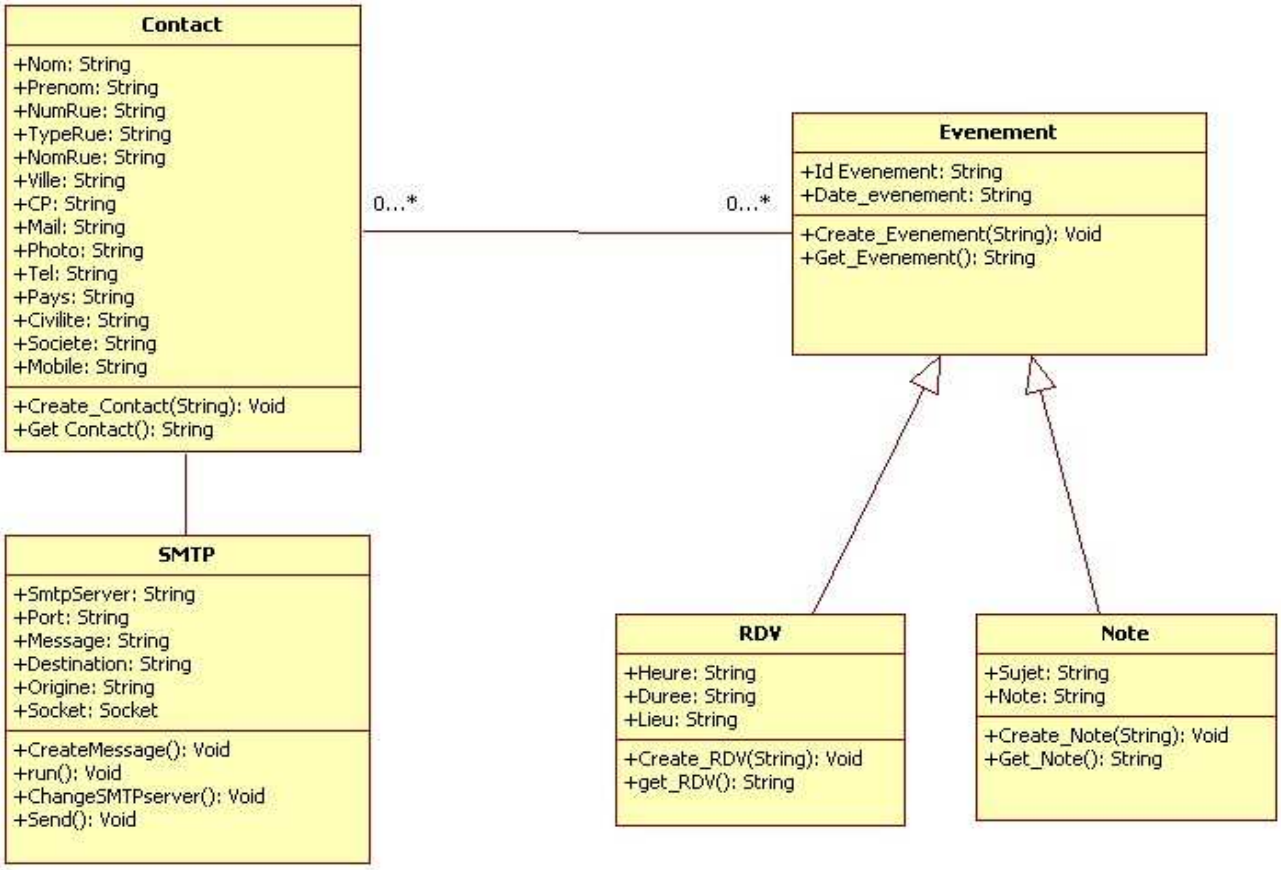
3.1 Diagramme des cas d'utilisation :

L'utilisateur doit s'authentifier pour pouvoir se servir de l'application , une fois authentifié , il peut ou il pourra consulter le calendrier, envoyer des mails , gérer ses rendez-vous , ses notes et ses contacts.



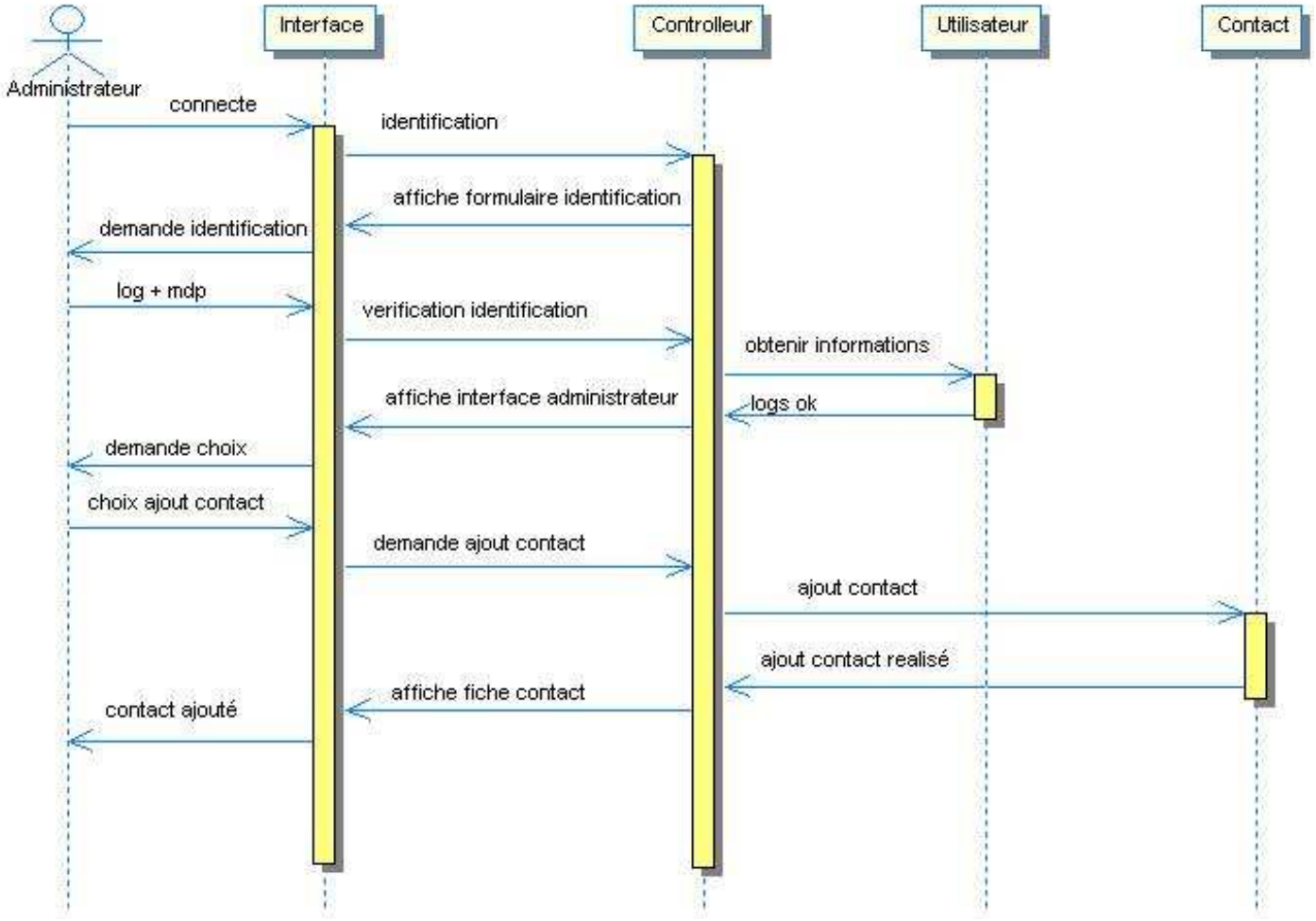
3.2 Diagramme de classe

Nous avons une classe contact, un événement est lié à un contact. Un contact à plusieurs natures, soit un rendez-vous, soit une note. Le client SMTP ira piocher dans les attributs de la classe contact pour récupérer l'adresse mail.

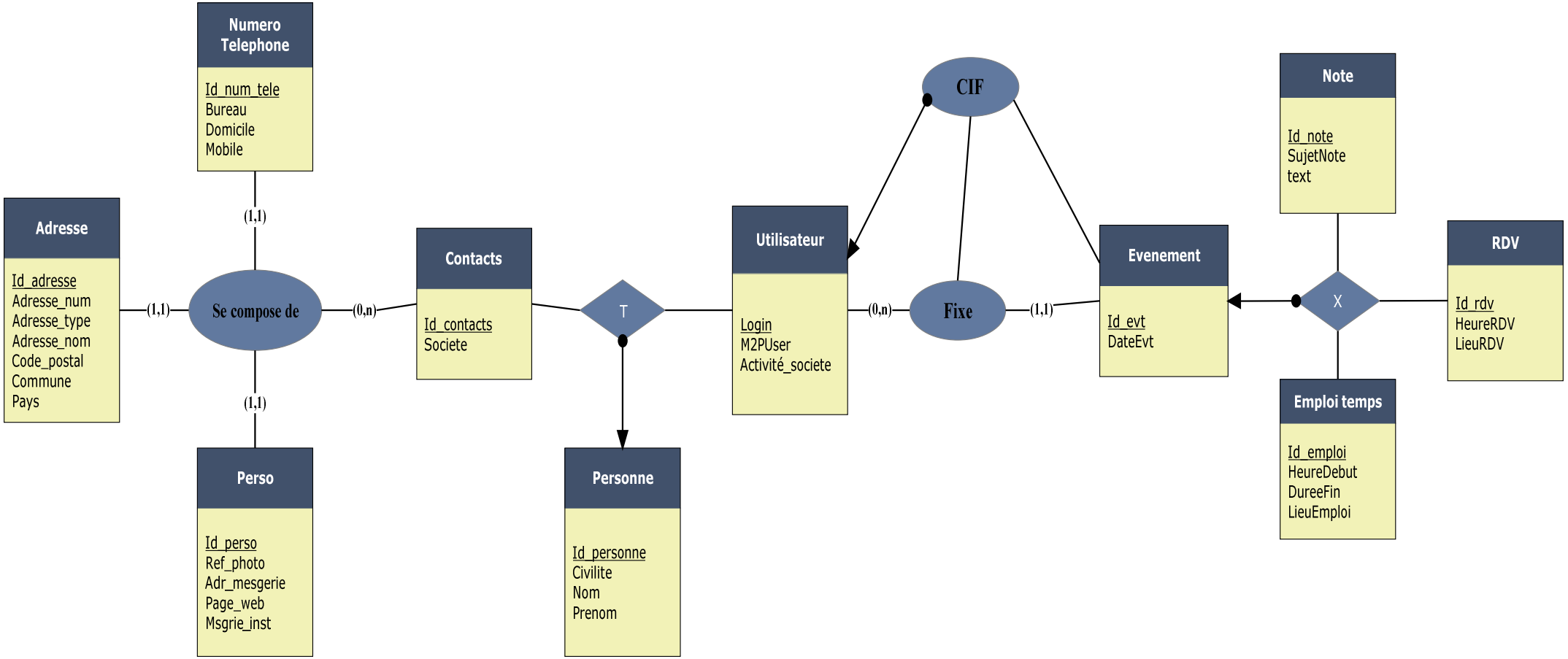


3.3 Diagramme de séquence :

Le client s'authentifie choisie d'enregistrer un contact, a ce moment la classe contrôleur cherche l'IHM correspondante à l'enregistrement d'un contact, l'utilisateur n'a plus qu'a remplir les champs et enregistrer, un objet contact est ensuite créer qui est envoyé à une classe LIFECYCLE s'occupant d'insérer le contact dans la base de données.



3.4 Model conceptuel de données :



3.5 Dictionnaire des données documenté :

Propriété	Signification
Mnémonique	Abréviation du nom de l'attribut
Libellé	Libellé contenant la signification précise et le rôle de l'attribut
Type de donnée	Type de l'attribut : entier, réel, chaîne de caractères, date...
Contraintes d'intégrité	Liste des contraintes sur les valeurs possibles de l'attribut

Table	Mnémonique		Libellé	Type	Contraintes
Adresse		Id_adresse	Identité unique attribué pour chaque l'adresse	Entier	Auto incrémente
	Fk_Id_contacts		Clé étrangère qui référence Id_contacts de la table Contacts	Chaîne(30)	ON UPDATE CASCADE ON DELETE CASCADE
	Adresse_num		Numéro de l'adresse	Entier	
	Adresse_type		Type de l'adresse	Chaîne(10)	
	Adresse_nom		Nom de l'adresse	Chaîne(30)	
	Code_postal		Le code postal	Entier(10)	
	Commune		Nom de la commune	Chaîne(20)	
	Pays		Pays	Chaîne(20)	

Numero_Telephone		Id_Num_tele	Identité unique attribué pour chaque Numero_Telephone	Entier(5)	Auto incrémente
	Fk_Id_contacts		Clé étrangère qui référence Id_contacts de la table Contacts	Chaîne(30)	ON UPDATE CASCADE ON DELETE CASCADE
	Bureau		Numero du bureau	Entier(20)	
	Domicile		Numero du domicile	Entier(20)	
	Mobile		Numero du portable	Entier(20)	
Perso		Id_perso	Identité unique	Entier(3)	Auto incrémente
	Fk_Id_contacts		Clé étrangère qui référence Id_contacts de la table Contacts	Chaîne(30)	ON UPDATE CASCADE ON DELETE CASCADE
	Ref_photo		Ref de la photo	Chaîne(50)	
	Adr_mesgrie		Adresse mail	Chaîne(30)	
	Page_web		Website	Chaîne(30)	
	Mesgrie_inst		Messagerie instantane comme MSN ou autre	Chaîne(30)	
Contacts		Id_contacts	Identité unique attribué pour chaque contacts	Chaîne(30)	Auto incrémente ??
	fk_Id_Personne		Clé étrangère qui référence Id_Personne de la table Personne	Chaîne(30)	ON UPDATE CASCADE ON DELETE CASCADE
	Societe		Nom de la société	Chaîne(20)	

Personne		Id_Personne	Identité unique attribué pour chaque personne	Chaîne(30)	Auto incrémente ???
	Civilite		M., Mme, Mlle	Chaîne(10)	
	Nom		Nom de la personne	Chaîne(30)	
	Prenom		Prénom de la personne	Chaîne(30)	
Evenement		Id_evt	Identité unique attribué pour chaque evenement	Entier(3)	Auto incrémente
	fk_Id_user		Clé étrangère qui référence Login de la table Utilisateur	Chaîne(10)	ON UPDATE CASCADE ON DELETE CASCADE
	DateEvt		Date	Date	
Note		Id_note	Identité unique attribué pour chaque note	Entier(3)	
	fk_Id_evt		Clé étrangère qui référence Id_evt de la table Evenement	Entier(3)	ON UPDATE CASCADE ON DELETE CASCADE
	SujetNote		Sujet de la note	Chaîne(20)	
	Text		Le contenu de note	Texte	
RDV		Id_rdv	Identité unique pour chaque rdv	Entier(3)	
	fk_Id_evt		Clé étrangère qui référence Id_evt de la table Evenement	Entier(3)	ON UPDATE CASCADE ON DELETE CASCADE

	HeureRDV		Heure de rdv	Time	
	LieuRDV		Lieu du rdv	Chaîne(50)	
Emploi_Temps		Id_emploi	Identité unique pour chaque emploi de temps	Entier(3)	
	fk_Id_evt		Clé étrangère qui référence Id_evt de la table Evenement	Entier(3)	ON UPDATE CASCADE ON DELETE CASCADE
	HeureDebut		Heure de debut	Time	
	HeureFin		Heure de fin	Time	
	LieuEmploi		Lieu du travail	Chaîne(50)	
Utilisateur		Login	Identité unique pour chaque utilisateur	Chaîne(10)	
	fk_Id_Personne		Clé étrangère qui référence Id_Personne de la table Personne	Chaîne(30)	ON UPDATE CASCADE ON DELETE CASCADE
	M2PUSER		Mots de passe utilisateur	Chaîne(10)	Not Null
	Activite_societe		L'activité de la société	Chaîne(30)	

3.6 Modèle Logique de Données :

Données à stocker:
Pour chaque table, on va lister les données à stocker dans cette base :

Adresse(Id_adresse, Adresse_num, Adresse_type, Adresse_nom, Code_postal, Commune, Pays, fk_Id_contacts)
fk_Id_contacts de Adresse fait référence à **Id_contacts** de Contacts
fk_Id_contacts de Adresse est obligatoire

Numero_Telephone(Id_Num_tele, Bureau, Domicile, Mobile, fax, fk_Id_contacts)
fk_Id_contacts de Numero_Telephone fait référence à **Id_contacts** de Contacts
fk_Id_contacts de Numero_Telephone est obligatoire

Perso(Id_perso, Ref_photo, Adr_mesgrie, Page_web, Mesgrie_inst, fk_Id_contacts)
fk_Id_contacts de Perso fait référence à **Id_contacts** de Contacts
fk_Id_contacts de Perso est obligatoire

Personne(Id_Personne, Civilite, Nom, Prenom)

Contacts(Id_contacts, Titre, Societe, Activite_societe, fk_Id_Personne)
Fk_Id_Personne de Contacts fait référence à **Id_Personne** de Personne

Utilisateur(Login, M2PUser, Activite_societe, fk_Id_Personne)
Fk_Id_Personne de Utilisateur fait référence à **Id_Personne** de Personne

Evenement(Id_evt, DateEvt, fk_Id_user)
fk_Id_user de Evenement fait référence à **Id_user** de Utilisateur
fk_Id_user de Evenement est obligatoire

Note(Id_note, SujetNote, Text, fk_Id_evt)
fk_Id_evt de Note fait référence à **Id_evt** de Evenement

RDV(Id_rdv, HeureRDV, LieuRDV, fk_Id_evt)
fk_Id_evt de RDV fait référence à **Id_evt** de Evenement

Emploi_Temps(Id_emploi, HeureDebut, HeureFin, LieuEmploi, fk_Id_evt)
fk_Id_evt de Emploi_Temps fait référence à **Id_evt** de Evenement

3.7 La mise en place de la base de données :

Toutes les requêtes de créations des tables et des contraintes ont été placé dans un fichier « .sql » en respectant l'ordre selon l'appartenance des clés étrangère.

4 LA REALISATION DU PROJET

4.1 Les problèmes rencontrés :

- Création d'une base de données via du code java !!!
  Pour pouvoir exécuter du code SQL sous java , il faut obligatoirement se connecter à une base de donnée mais si le code SQL à exécuter en l'occurrence est une requête concernant la création d'une base de données , il a fallu pour cela trouver une base de donnée qui soit d'emblée présent dans la solution wamp ou autre (ex : MySQL), il se trouve qu'une base de donnée nommée MySQL y est présente.
  Donc nous nous connectons à cette base et créons une base de données propre à chaque utilisateur pour gérer ses contacts.

```
public void Create_Database() throws SQLException, ClassNotFoundException, IOException
{
    if(Verify_DB())
    {
        Class.forName("com.mysql.jdbc.Driver");
        String url = "jdbc:mysql://localhost:3306/mysql";
        String user = "root";
        String pass = "";
        // on commence par se connecter à la base factice
        Connection factice = DriverManager.getConnection(url,user,pass);
        // on crée la base et on récupère une Connection
        Connection connection = createMysqlDatabase(factice,user,pass,this.Field_login.getText());
        // on peut finalement fermer notre Connection factice qui ne nous sers plus à rien
        factice.close();
        connection.close();
        // on ferme la bonne connection avec la nouvelle base de données
        connection.close();
        structure_database(this.Field_login.getText());
    }
}
```

```
public static Connection createMysqlDatabase(Connection factice, String user,String pass, String nomBase) throws SQLException
{
       Connection connection = null;
       Statement statement = null;
       try
       {
          statement = factice.createStatement();
          statement.execute("CREATE DATABASE "+nomBase);
          String url = factice.getMetaData().getURL();
          url = url.substring(0,url.lastIndexOf("/"));
          url += "/"+nomBase;
          connection = DriverManager.getConnection(url,user,pass);
       }
       catch(SQLException e)
       {
          SQLException sqle = new SQLException("Création de la base impossible");
          sqle.setNextException(e);
          throw sqle;
       }
       finally
       {
          try{statement.close();}catch(Exception e){}
       }
       return connection;
}
```

- Une IHM java n'est pas faite pour accueillir d'image de fond

  Nous avons voulue dans notre application mettre une image en fond, hors en java seul le Label convient à l'insertion d'image, il a donc fallu créer un label, pour pouvoir par la suite faire une sorte de « dérivation » de la frame pour qu'elle récupère les propriétés du label

```
AuthentificationIHM fenetre_log = new AuthentificationIHM(); // frame d'authentification
/* Charge l'image en fonde de la jframe */
IHMFond panel = new IHMFond(new ImageIcon("src\\stag\\icone\\fond.jpg").getImage());
fenetre_log.getContentPane().add(panel);
```

```
    public IHMFond(Image img)
    {
        this.img = img;
        Dimension size = new Dimension(img.getWidth(null), img.getHeight(null));
        setPreferredSize(size);
        setMinimumSize(size);
        setMaximumSize(size);
        setSize(size);
        setLayout(null);
    }

    @Override
    public void paintComponent(Graphics g)
    {
        g.drawImage(img, 0, 0,null);
    }
```

4.2 Les différentes classes :

4.2.1 IHM :

- ContactIHM : Enregistrement de contact
- AuthentificationIHM : Fenêtre d'authentification
- ListeContactIHM : Listes les contactes enregistré dans la base de données
- CréationLoginIHM : Pour la création des comptes utilisateurs
- MenuIHM : Interface contenant les différents menus

4.2.2 ENTITY :

- Contact : Classe contenant tous les attributs d'un contact
- SMTP : Classe qui gère l'envoi de mail

4.2.3 CONTROLEUR :

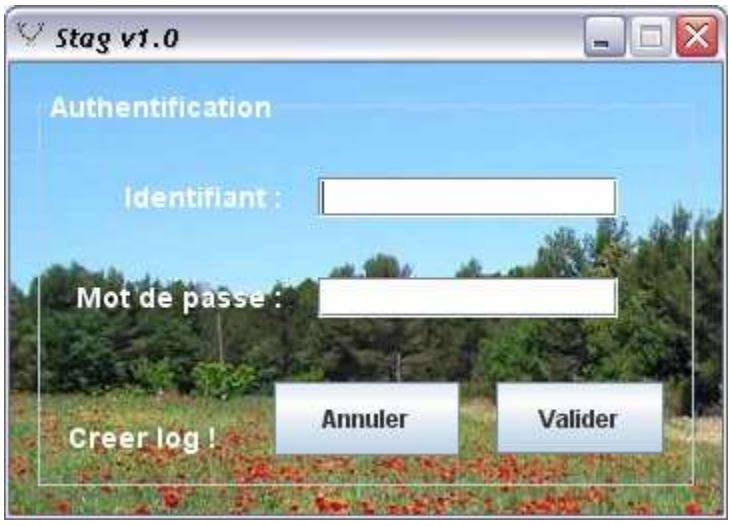
- Classe contrôleur qui fait le lien entre toutes les classes

4.2.4 LIFECYCLE :

- Gestion_DB : Connexion, sélection et enregistrement dans la base de données

4.3 Les différents menus et fonctionnalités:

4.3.1 Interface d'authentification :

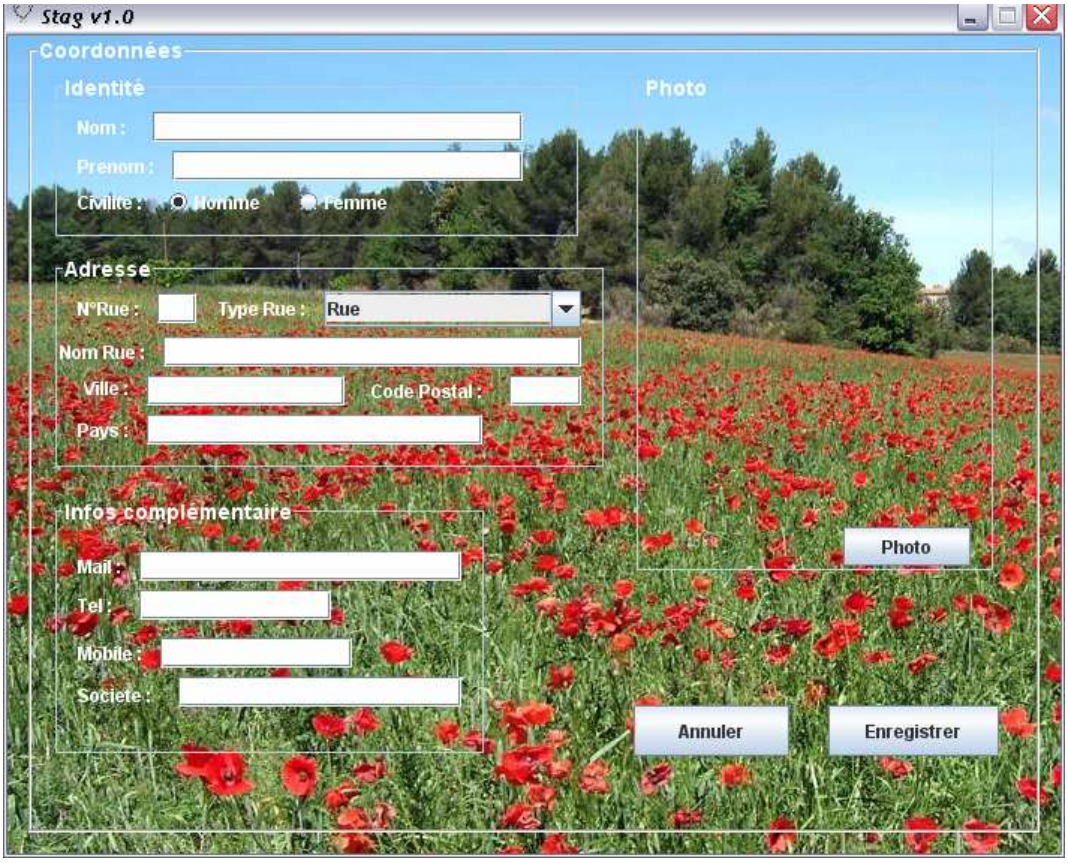


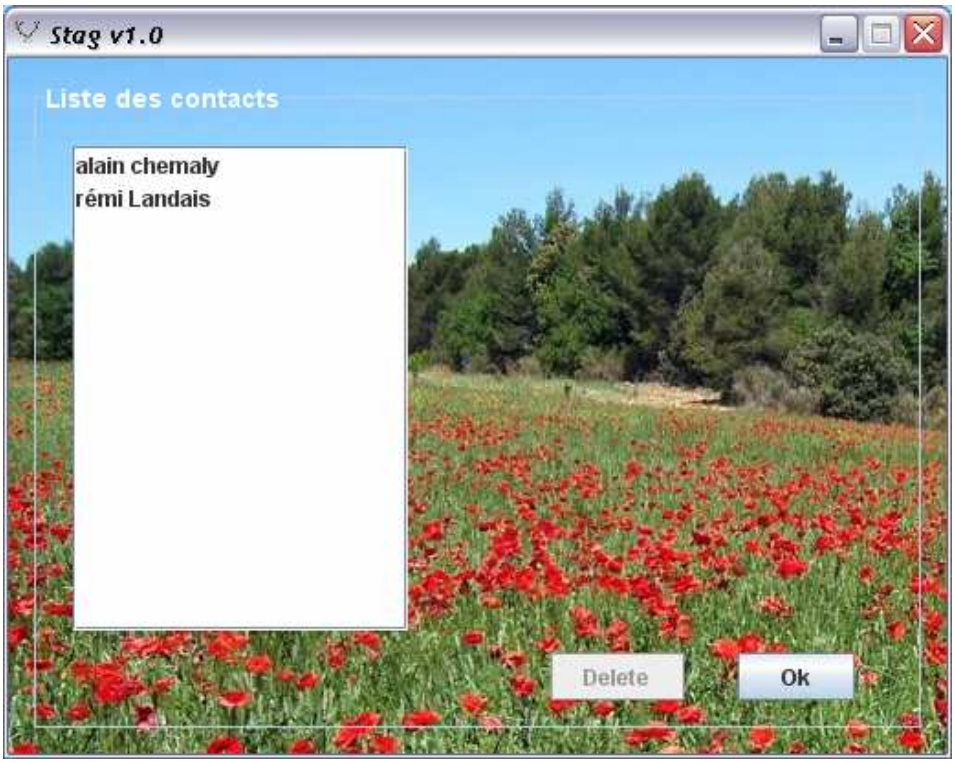
4.4 Interfaces principales:

Dans l'interface principale on a le Menu composé de trois modules :

- Création d'un contact
- Modification d'un contact
- Calendrier

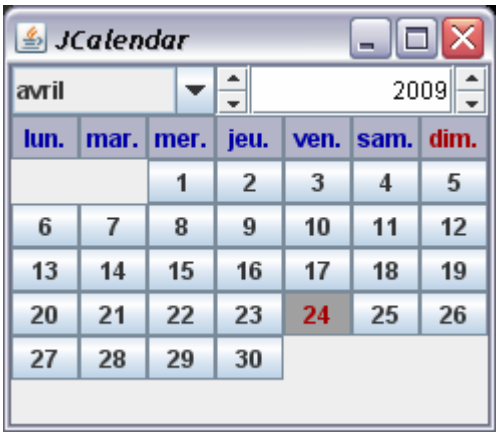
Dans création d'un compte, on peut introduire différentes information sur la personne comme une photo, son adresse…





On a la possibilité aussi de modifier tous les contacts insérés préalablement en sélectionnent dans les listes des contacts.

Et puis le module calendrier, qui nous permet de visualiser les dates du mois et la prise des notes et prendre des rendez vous.



Alain CHEMALY et Rémi LANDAIS

Les Exceptions :

```
try
{
    /* Connection à la base de données STAG avec log : root et pwd : 090988 */
    connexion = DriverManager.getConnection("jdbc:mysql://localhost:3306/"+login, "root", dbpassword);

} catch (SQLException sQLException)
{
    /* affichage message d'erreur */
    JOptionPane.showMessageDialog(null, "Erreur sur la connexion à la base de donnée !");
}
```

Ici nous créons la connexion avec la base concernant à l'utilisateur authentifié, si une erreur SQL survient le code dans le catch est appelé, l'exception est attrapé. En cas d'exception de type SQL nous affichons une boite de dialogue indiquant qu'il y a eu un problème lors de la connexion à la base de données

Surcharge de méthode:

Ici une frame de type ContactIHM est pris en paramètre

```
/* fonction servant à mettre l'icone */
public static void Set_icone(String path, ContactIHM frame)
{
    File icone = new File(path);
    if(icone.exists())
    {
         ImageIcon icon = new ImageIcon(path);
         frame.setIconImage(icon.getImage());
    }
    else
    {
        /* affichage message d'erreur */
        JOptionPane.showMessageDialog(null, "Erreur de chargement sur l'icone de la frame !");
    }

}
```

Et plus loin c'est une frame de type CreationLogIHM

```
  public static void Set_icone(String path, CreationLogIHM frame)
{
    File icone = new File(path);
    if(icone.exists())
    {
         ImageIcon icon = new ImageIcon(path);
         frame.setIconImage(icon.getImage());
    }
    else
    {
        /* affichage message d'erreur */
        JOptionPane.showMessageDialog(null, "Erreur de chargement sur l'icone de la frame !");
    }

}
```

Ces méthodes ont pour toutes les deux la fonction de changer l'icône de la frame mais ne vise pas le même type de frame.

Travail Restant:

- Il ne manque plus que le client smtp à finir
- Module création de notes et de rendez-vous
- Gestion d'alertes pour les rendez-vous