Architecture

Plateforme de Réseaux sociaux combinés

Architecture

Plateforme de Réseaux sociaux combinés

Introduction **3**

I. L'architecture JEE **3**

a. Architecture du projet et diagramme de classe....3

b. Nature des pages et package....6

II. Architecture et Base De Données **6**

a. Schéma global....6

b. Les données stockées en Base De Données....7

Introduction

L'objectif de cette application Web est de centraliser l'ensemble des réseaux sociaux d'un utilisateur (première Facebook puis Twitter) sur une seule et même application : Pluss. Et pouvoir à partir de cette application Web, publier sur l'ensemble des réseaux sociaux voulus de l'utilisateur d'un seul clic de souris. La fonctionnalité secondaire est de récupérer l'ensemble des flux voulus des divers réseaux sociaux de l'utilisateurs sur la page d'accueil du site Web Pluss.

I. L'architecture JEE

a. Architecture du projet et diagramme de classe

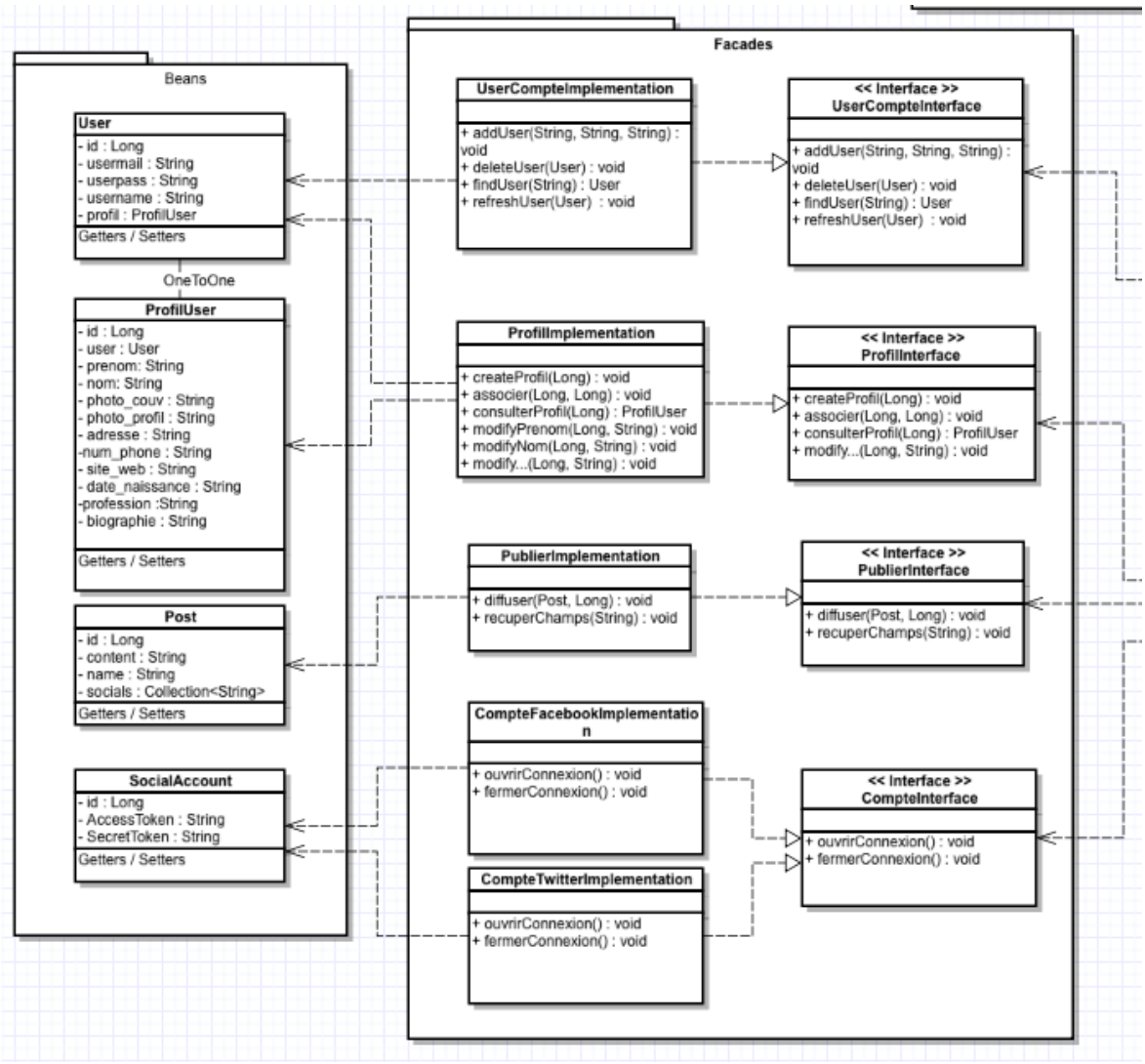


figure 1 *Diagramme de classe : partie 1*

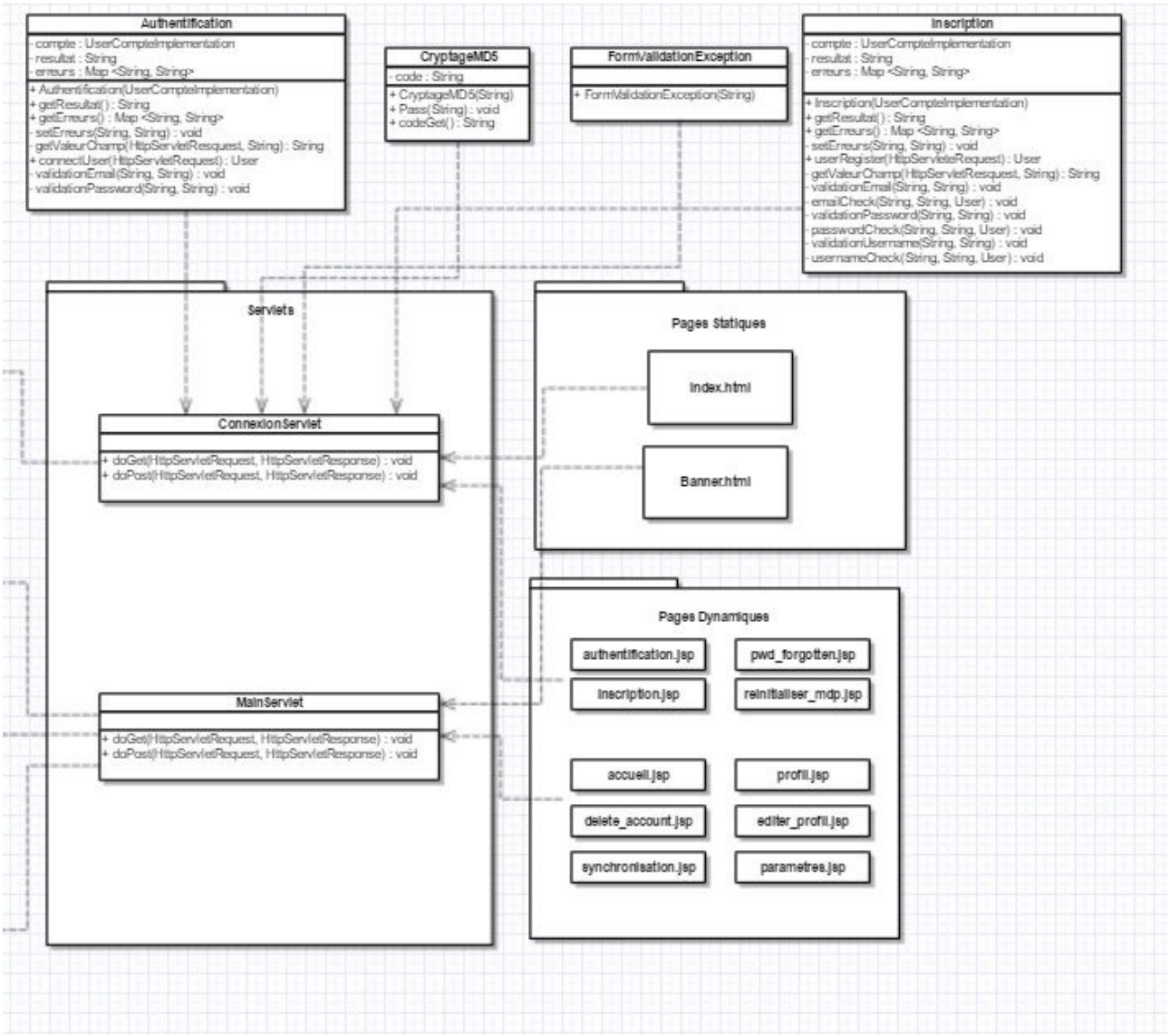
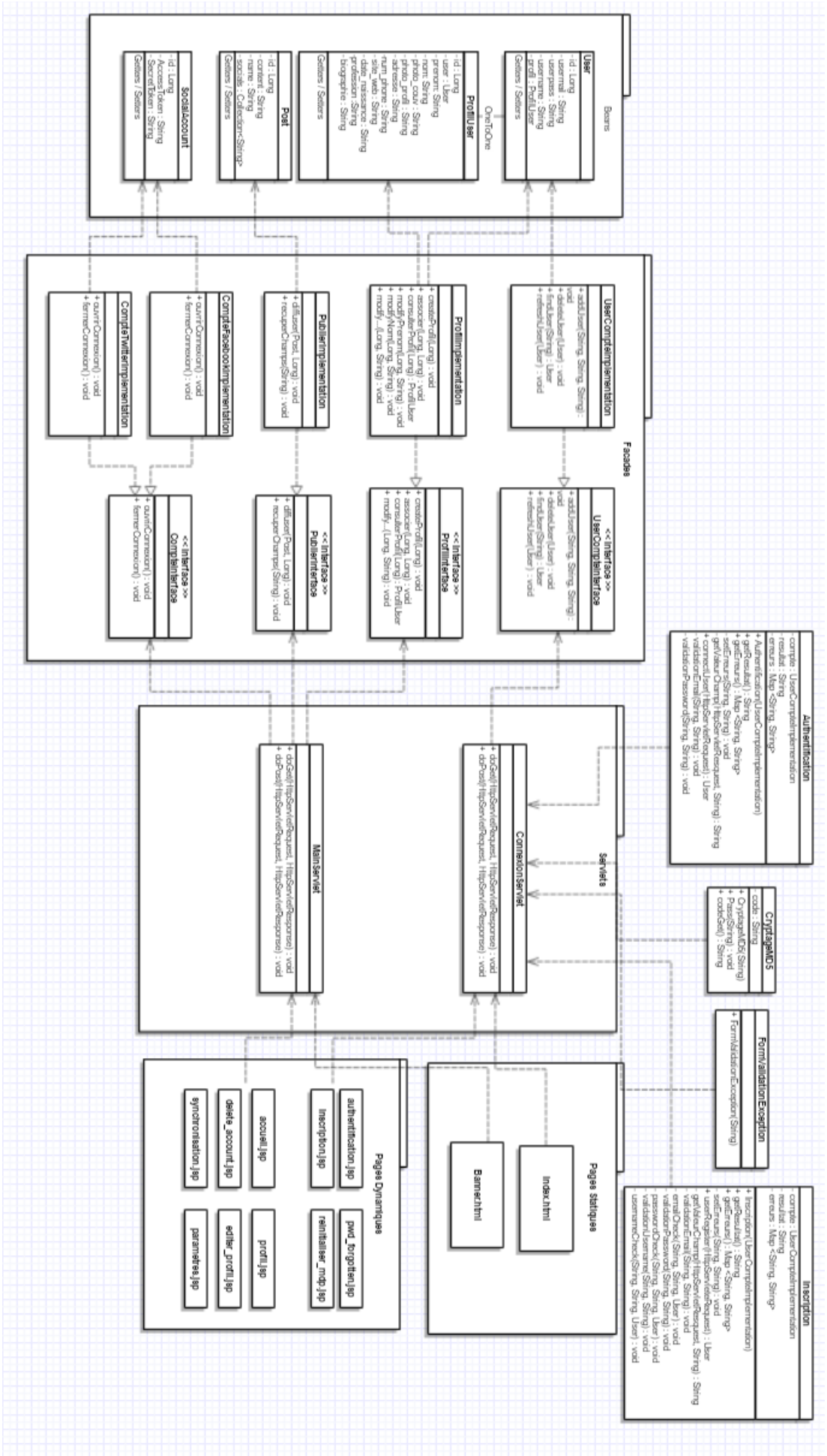


figure 2 *Diagramme de classe : partie 2*

figure 3 *Diagramme de classe : COMPLET*

Note : les classes Authentification, Inscription, CryptageMD5 et FormInvalidException appartiennent au package form contenant l'ensemble des classes utilitaires utiles à la servlet ConnexionServlet.


Beans
User
ProfilUser
Post
SocialAccount
OneToOne
Facades
UserComptaImplementation
ProfilImplementation
PublierImplementation
CompteFacebookImplementation
CompteTwitterImplementation
<< Interface >>
UserComptaInterface
<< Interface >>
ProfilInterface
<< Interface >>
PublierInterface
<< Interface >>
ComptaInterface
Authentification
Inscription
CryptageMD5
FormValidationException
Servlets
ConnexionServlet
MainServlet
Pages Statiques
Banner.html
Index.html
Pages Dynamiques
authentification.jsp
Inscription.jsp
reinitialiser_mdp.jsp
pwd_forgotten.jsp
accueil.jsp
profil.jsp
delete_account.jsp
editer_profil.jsp
synchronisation.jsp
parametres.jsp

b. Nature des pages et package

Comme indiqué sur le schéma précédent, on retrouve :

- page statique : index.html (authentification)
- page dynamique : le reste, la bannière comprise. Laissée en html pour le moment et sur le diagramme car difficulté à l'inclure dans une autre page jsp, étant elle même une jsp.
- servlets : MainServlet, ConnexionServlet
- façades : UserCompteImplementation héritant de UserCompteInterface, UserProfilImplementation héritant de UserProfilInterface, PublierImplementation héritant de PublierInterface, CompteFacebookImplementation héritant de CompteInterface, CompteTwitterImplementation héritant de CompteInterface.

II. Architecture et Base De Données

a. Schéma global

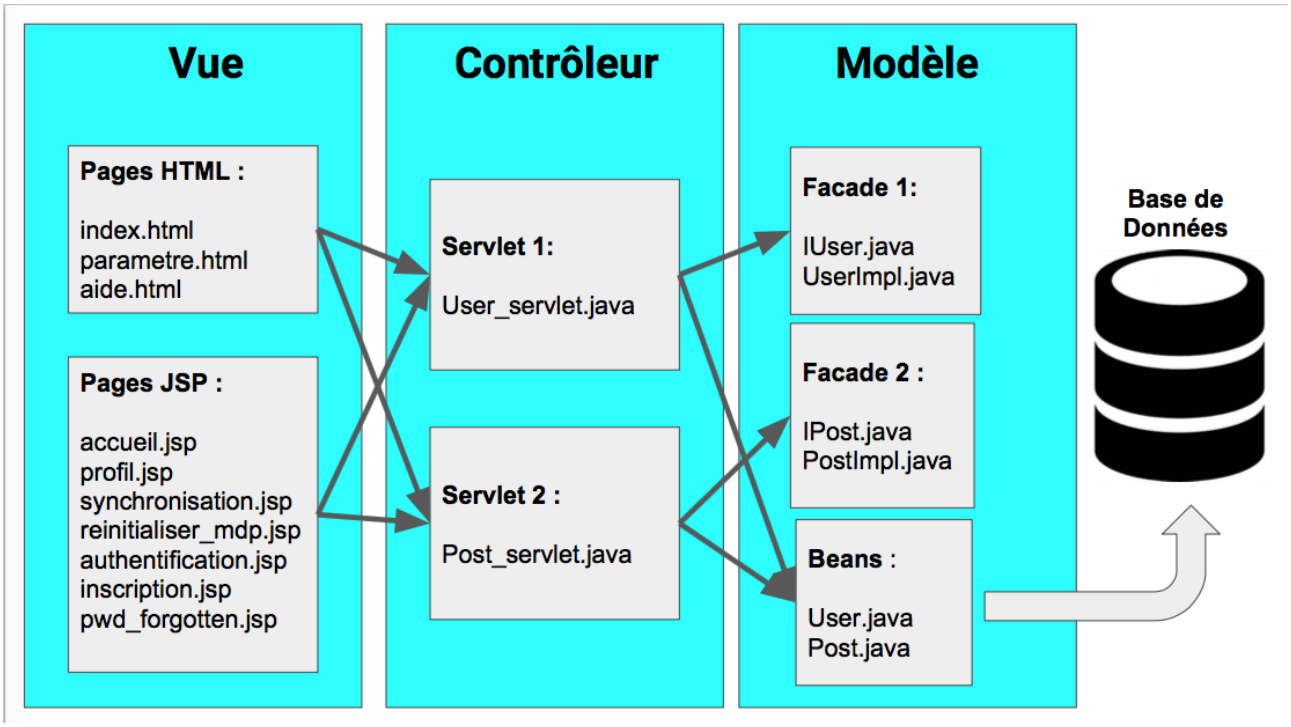


figure *Schéma MVC*

b. Les données stockées en Base De Données

```
@Entity
public class User {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long   id;
    private String usermail;
    private String userpass;
    private String username;
    //private Timestamp registerDate;

    @OneToOne
    private ProfilUser profil;   //Axel

    public ProfilUser getProfil() { //Axel
        return profil;
    }
    public void setProfil(ProfilUser profil) { //Axel
        this.profil = profil;
    }


    public Long getId() {
        return id;
    }
    public void setId( Long id ) {
        this.id = id;
    }

    public void setEmail(String email) {
    this.usermail = email;
    }
    public String getEmail() {
    return usermail;
    }

    public void setPassword(String password) {
    this.userpass = password;
    }
    public String getPassword() {
    return userpass;
    }

    public void setUsername(String username) {
    this.username = username;
    }
    public String getUsername() {
```

figure *L'utilisateur : User*

```java
@Entity
public class ProfilUser {

    @Id
    private long id;

    @OneToOne(mappedBy="profil",fetch = FetchType.EAGER)
    private User user;

    private String prenom;
    private String nom;
    private String photo_couv;
    private String photo_profil;
    private String adresse;
    private String num_phone;
    private String site_web;
    private String date_naissance;
    private String profession;
    private String biographie;

    //---------  getters  ----------//

    public Long getId() {
        return id;
    }
    public User getUser() {
        return user;
    }
    public String getPrenom() {
        return prenom;
    }
    public String getNom() {
        return nom;
    }
    public String getPhotoCouv() {
        return photo_couv;
    }
    public String getPhotoProfil() {
        return photo_profil;
    }
    public String getAdresse() {
        return adresse;
    }
    public String getNumPhone() {
        return num_phone;
    }
    public String getSiteWeb() {
        return site_web;
    }
    public String getDDN() {
        return date_naissance;
```

figure *Le profil de l'utilisateur : ProfilUser*

```
3   @Entity
4   public class SocialAccount {
5
6       @Id
7       @GeneratedValue(strategy=GenerationType.AUTO)
8       private Long   id;
9       private String AccessToken;
10      private String SecretToken;
11
12  }
13
```

figure *Les comptes réseaux sociaux des utilisateur : SocialAccount*