

AppliBump, guide du programmeur

PROJET INDIVIDUEL

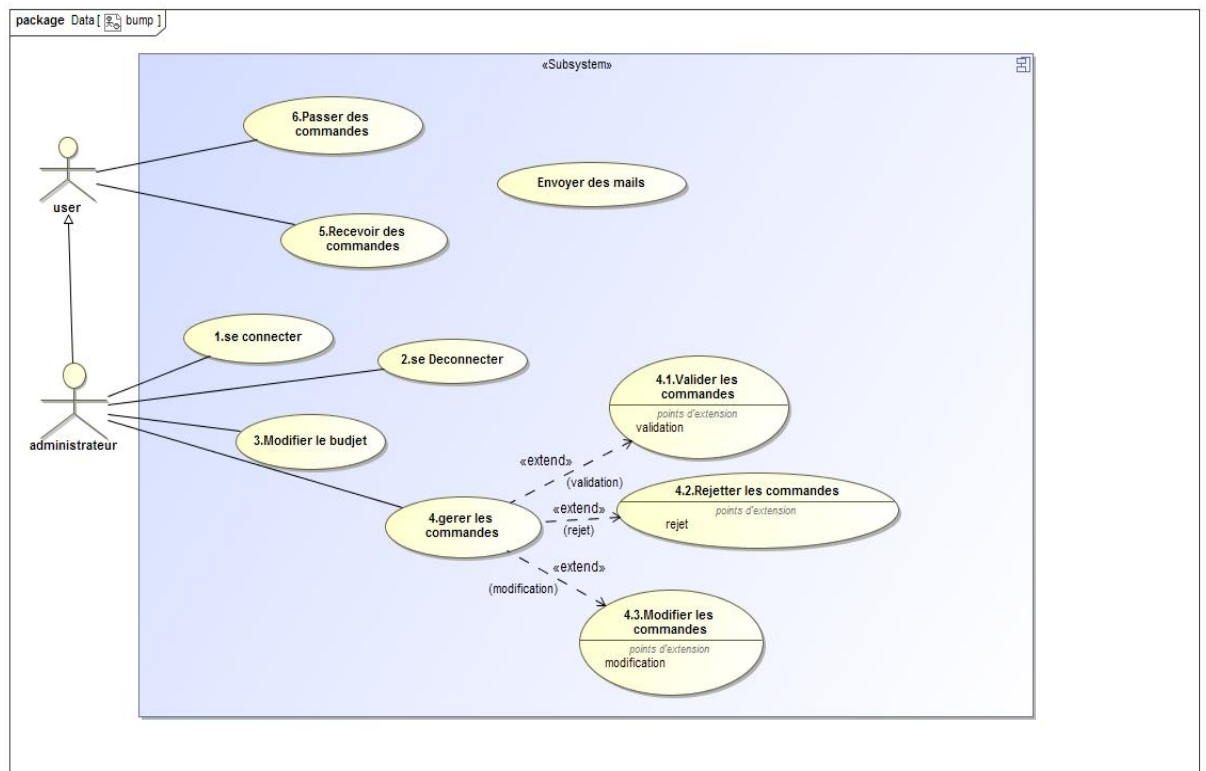
Marie Ntumb An Mwinkeu
UNIVERSITE DE NAMUR |

Tables de matières

1. Modélisation
2. Gestion des formulaires
3. L'accès à la base de données
4. Paramètres mail
5. Gestion de l'affichage
6. Mise en page
7. Createdatabase

1. Modélisation

Notre modélisation s'est effectuée en deux étapes. Premièrement nous avons élaboré un Uses cases afin de mettre en évidence les relations fonctionnelles entre les acteurs (client, administrateur) et le système étudié.



Et deuxièmement, nous avons conçu un diagramme de séquence afin d'illustrer le uses cases ci-haut, tout en y mettant l'accent sur la chronologie des envois des messages.



2. Gestion des formulaires

Tous les champs du formulaire, en plus de deux valeurs entières sont encapsulés dans le Bean nommé : « monformulaire.java » se trouvant dans le package « Beans ». La première valeur « **numerocli** », est l'identifiant de la personne qui passe la commande. Et la deuxième valeur, « **numerocom** », est un numéro qu'on attribue à la commande, elle sera utile lors des différents traitements que nous effectuerons au sein de notre application tels que la sauvegarde de la commande dans une base des données, la modification de la commande, la validation de la commande ainsi que la suppression de la commande ou encore voir toutes les commandes passées. Ces deux valeurs sont générées automatiquement lors de l'enregistrement de la commande dans le bas de données.

3. L'accès à la base de données

Compte tenu de différentes fonctionnalités de notre application, nous utilisons la base de données pour garder une trace sur toutes les commandes passées afin de permettre une bonne gestion. Notre application se connecte à la base de donnée via la méthode « connectmy() » de la classe « Connexion.java » du package « manipulationbd ». Toute modification concernant l'accès à la base de données doit se faire à ce niveau.

```
public class Connexion {
    /**
     * connexion à la base de donnée
     * @return statut de la connexion
     */

    public static java.sql.Connection connectMy(){
        java.sql.Connection connexion=null ;
        try{
            Class.forName("org.h2.Driver");
            connexion = DriverManager.getConnection("jdbc:h2:tcp://localhost/~ /test", "sa","");
            /*System.out.println("hello");*/
        }catch(ClassNotFoundException e){}/*{System.out.print("erreur du driver");}*/
        catch(SQLException e){}/*{System.out.print("erreur de connection");}*/
        return connexion;
    }
}
```

Ce dernier package contient toutes les classes qui interagissent avec la base de données. Et les requêtes SQL plus complexes sont dans la classe « Requete » du package « requetesetmessage ».

- Structure de la base de données

4. Paramètres mail

La classe « Email » du package « Mail », avec sa méthode « sendemail » gère tout ce qui concerne l'envoi des mails. Le smtp utilisé est celui de Gmail.

- Adresse expéditeur : bumpprojet@gmail.com mot de passe : Bump2015

- Contenu et objet des mails : lors de la validation des commandes, ou des nouvelles commandes, différents mails sont envoyés soit au client (personne ayant passé une commande) soit au responsable. Le contenu ainsi que l'objet des messages sont modifiable à partir de la classe « Requete » du package « requetesetmessage »

5. Gestion de l'affichage

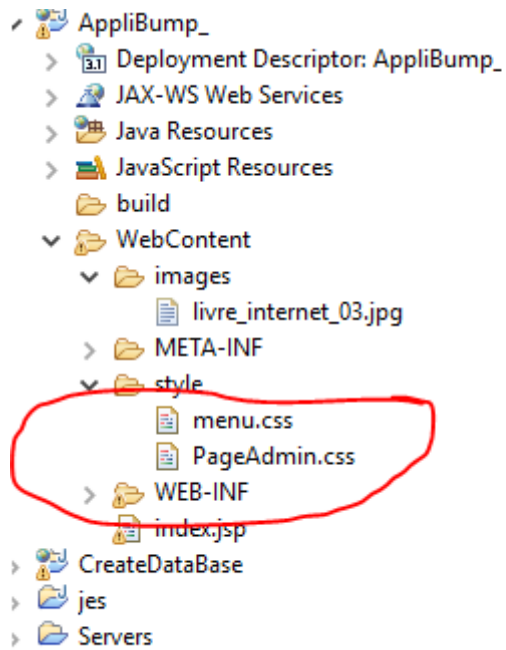
L'intérêt est de rendre possible la création de pages dynamiques car il y a une étape de génération sur le serveur, et l'affichage varie en fonction notamment de la requête et des données reçues.

Le tableau ci-dessous reprend les noms des différentes pages jsp ainsi que le type de contenu qu'elles affichent ou leurs rôles:

	Fichier JSP responsable
Le menu principal	Index.jsp
Le nouveau formulaire à remplir	Newformulaire.jsp
Les données remplies sur le formulaire par le client	Resultat.jsp
Le menu de l'administrateur	Accueiladmin.jsp
Modifier le budget	Budget2.jsp
Les erreurs susceptibles d'arriver lors d'une recherche quelconque	Erreur.jsp
Les nouvelles commandes	Commande1.jsp
Les commandes en cours ou déjà livrées	Commande2.jsp
La synthèse d'un livre	Livre.jsp
Modifier une commande	Modifierformulaire.jsp
Modifier les identifiants de l'administrateur	Mofieridentifiant.jsp
Réceptionner un livre	Receptionlivre.jsp
La synthèse d'un utilisateur	Utilisateur.jsp

6. Mise en page

Toutes les modifications concernant la mise en page doit être fait à partir des fichiers « menu.css » et « pageadmin.css ».



7. Createdatabase

C'est une petite application auxiliaire qui permet :

- ❖ De créer la base de données, qui sera utilisé par applibump ;
- ❖ D'initialiser les paramètres de connexion de l'administrateur.