Documentation technique logiciel Java

Projet Stephi Place Real Estate

Table des matières

[1. Besoins et contraintes 3](#_Toc41984714)

[1.1 Besoins 3](#_Toc41984715)

[1.2 Contraintes 3](#_Toc41984716)

[1.3 Conclusion 3](#_Toc41984717)

[2. Fonctionnalités de l’application 3](#_Toc41984718)

[2.1 Ce qui a vraiment été réalisé 3](#_Toc41984719)

[2.2 Diagramme de cas d’utilisation 4](#_Toc41984720)

[2.3 Diagrammes de séquence 4](#_Toc41984721)

[Connexion 4](#_Toc41984722)

[Affichage de la liste de clients 5](#_Toc41984723)

[Création d’un client 5](#_Toc41984724)

[Modification d’un client 6](#_Toc41984725)

[Suppression d’un client 6](#_Toc41984726)

[Création d’un bien 7](#_Toc41984727)

[Modification d’un bien 7](#_Toc41984728)

[Suppression d’un bien 8](#_Toc41984729)

[Création d’un type de bien 8](#_Toc41984730)

[Modification d’un type de bien 9](#_Toc41984731)

[Suppression d’un type de bien 9](#_Toc41984732)

[Création des images d’un bien 10](#_Toc41984733)

[Suppression des images d’un bien 10](#_Toc41984734)

[Création d’une vente 11](#_Toc41984735)

[Modification d’une vente 11](#_Toc41984736)

[Suppression d’une vente 12](#_Toc41984737)

[2.4 Description des données 12](#_Toc41984738)

[3. Choix technologiques 13](#_Toc41984739)

[3.1 Langage 13](#_Toc41984740)

[3.2 Logiciels utilisés 13](#_Toc41984741)

[3.3 Règles de codage et convention de nommage 14](#_Toc41984742)

[4. Design de l’application 14](#_Toc41984743)

[5. Liens utiles 15](#_Toc41984744)

[5.1 Liens des logiciels utilisées 15](#_Toc41984745)

[5.2 Lien vers le GitHub du projet 15](#_Toc41984746)

# Besoins et contraintes

## Besoins

Nous devons définir les besoins de l’application afin de pouvoir mieux appréhender les fonctionnalités voulues pour l’application.

L’application doit répondre à ces besoins :

-Gestion des clients (création, modification, suppression)

-Gestion des biens (création, modification, suppression)

-Tableau de bord agent immobilier

-Espace d’administration des agents et agences

-Gestion des annonces (création, modification, suppression)

-Gestion des documents (création, modification, suppression)

-Gestion des propositions d’achats/négociation

-Simulateur du processus de proposition/contre-proposition

## Contraintes

Pour la réalisation de ce projet, il nous a été demandé de respecter certaines contraintes qui sont les suivantes :

* Développement en Java
* Design Pattern: MVC
* Base de données commune avec Stephi Place Web Sales.

Ces contraintes devront être mises en œuvre pour la réalisation de ce projet.

## Conclusion

Grâce à l’étude des besoins de l’application ainsi que des contraintes à respecter, nous pouvons passer à la création de l’application.

# Fonctionnalités de l’application

## 2.1 Ce qui a vraiment été réalisé

Certains problèmes au cours du projet nous ont poussé à ne pas réaliser toutes les fonctionnalités prévues à la base.

Voici donc la liste de celles qui ont été réalisées et celles qui ne le sont pas :

-Gestion des clients (création, modification, suppression) - Réalisée

-Gestion des biens (création, modification, suppression) – réalisée

-Tableau de bord agent immobilier – non réalisée

-Espace d’administration des agents et agences – non réalisée

-Gestion des annonces (création, modification, suppression) – Réalisée

-Gestion des documents (création, modification, suppression) – Réalisée

-Gestion des propositions d’achats/négociation – Réalisée

-Simulateur du processus de proposition/contre-proposition – Réalisée

## 2.2 Diagramme de cas d’utilisation

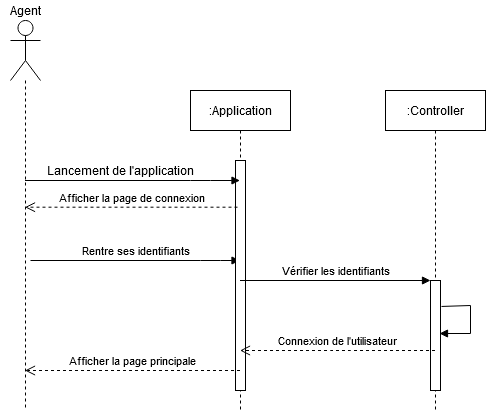
Le diagramme de cas d’utilisation représente les fonctionnalités de l’application du point de vue de l’utilisateur extérieur. Ce diagramme a été réalisé avec les informations obtenues dans la partie besoins :

IMAGE ICI

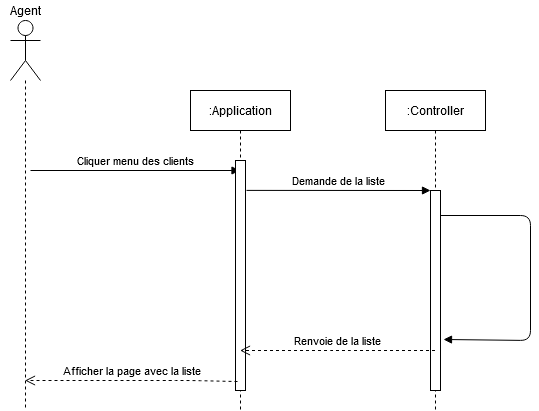
## 2.3 Diagrammes de séquence

Les diagrammes de séquence permettent de représenter les actions entre un utilisateur et le système en représentant les actions effectuées.

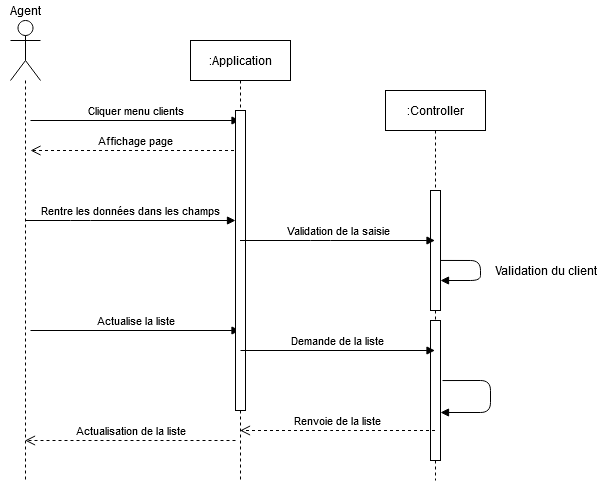
### Connexion



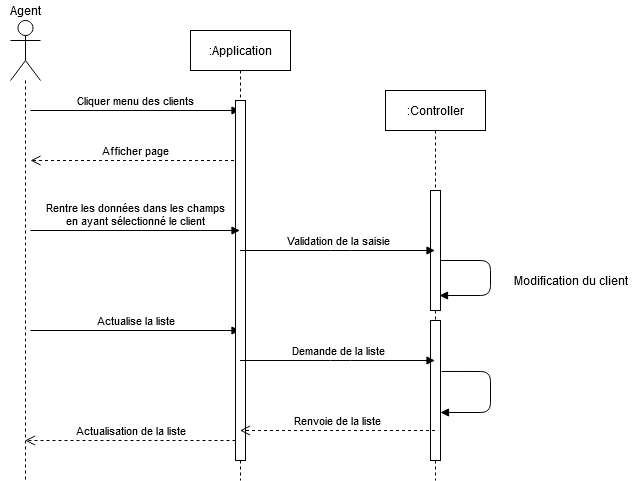
### Affichage de la liste de clients



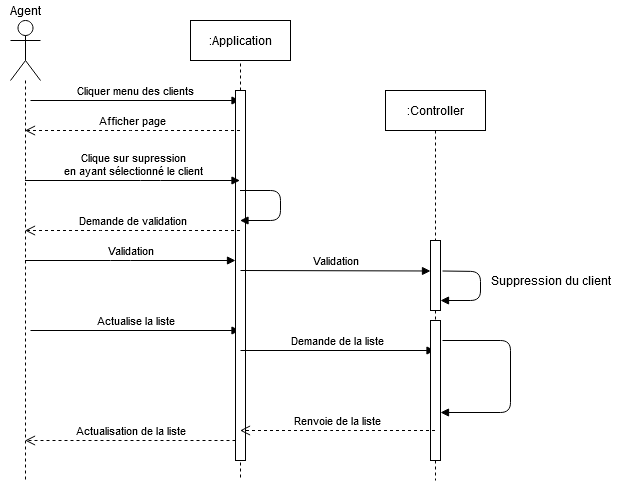
### Création d’un client



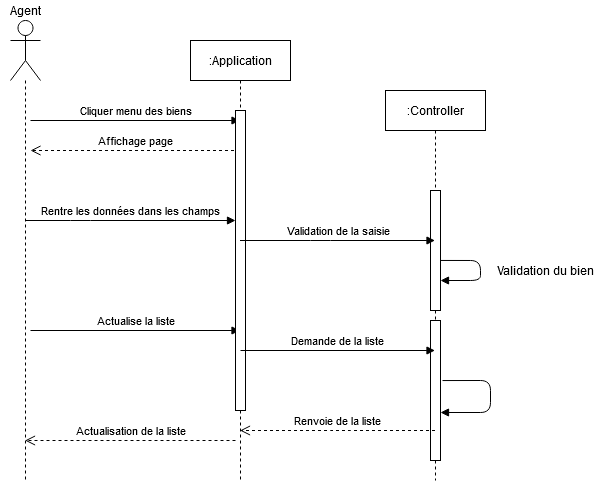
### Modification d’un client



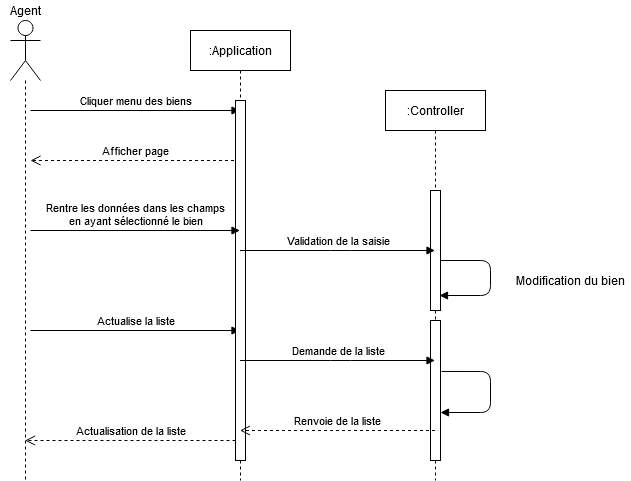
### Suppression d’un client



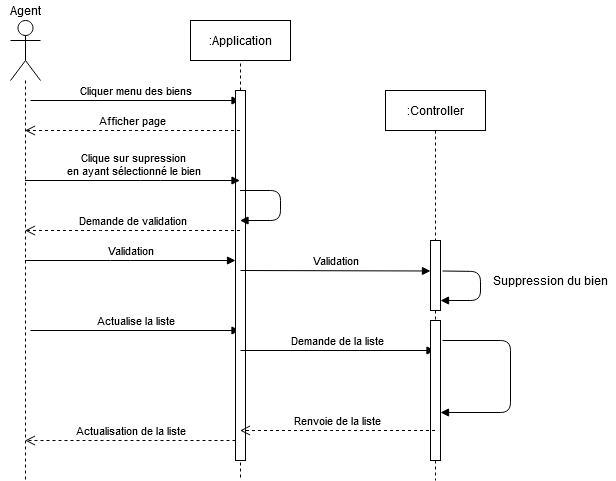
### Création d’un bien



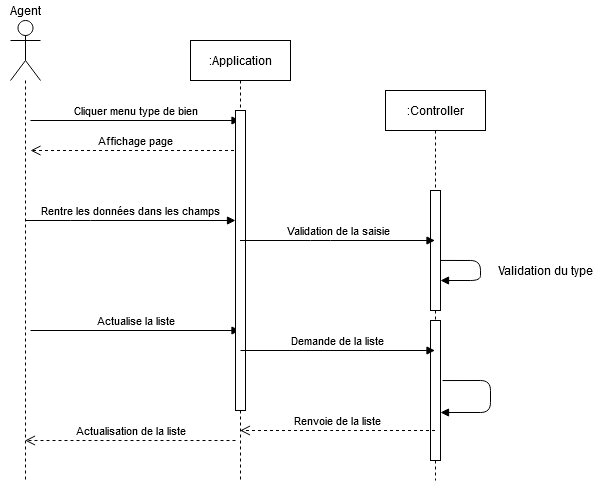
### Modification d’un bien



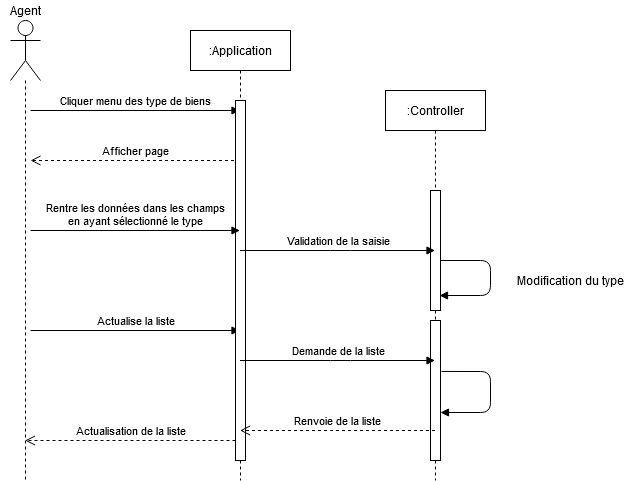
### Suppression d’un bien



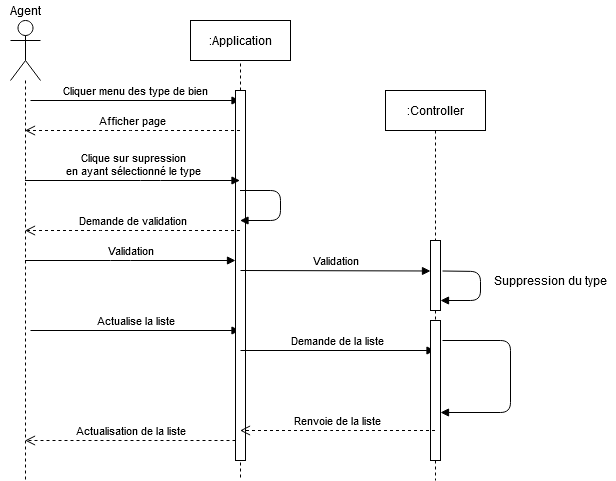
### Création d’un type de bien



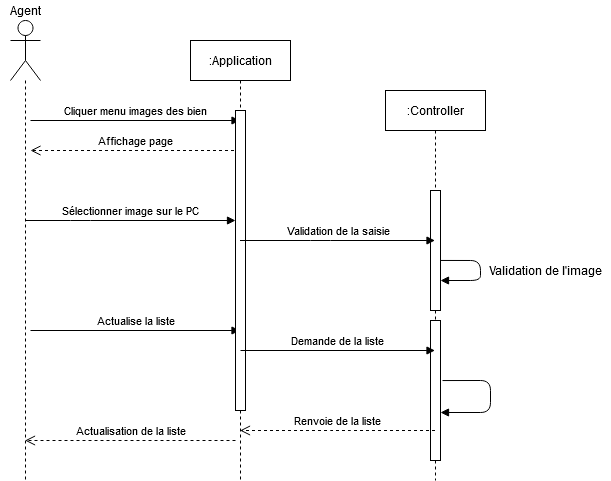
### Modification d’un type de bien



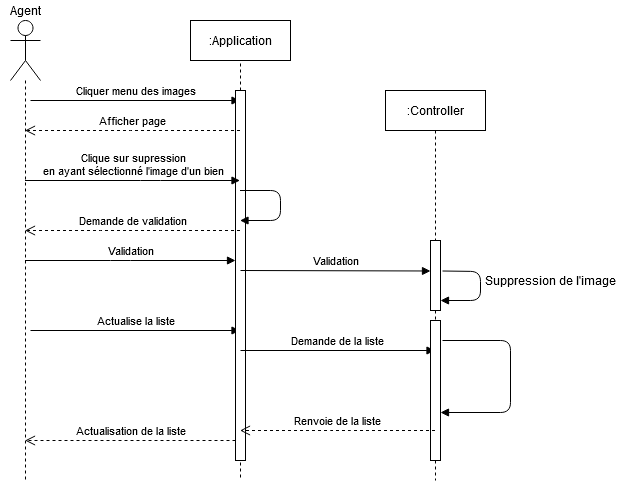
### Suppression d’un type de bien



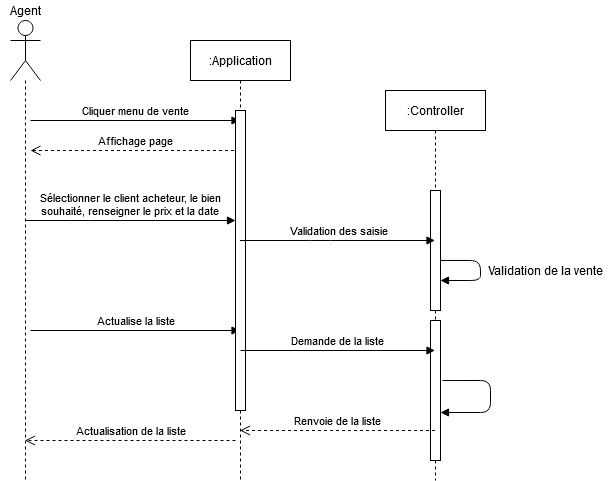
### Création des images d’un bien



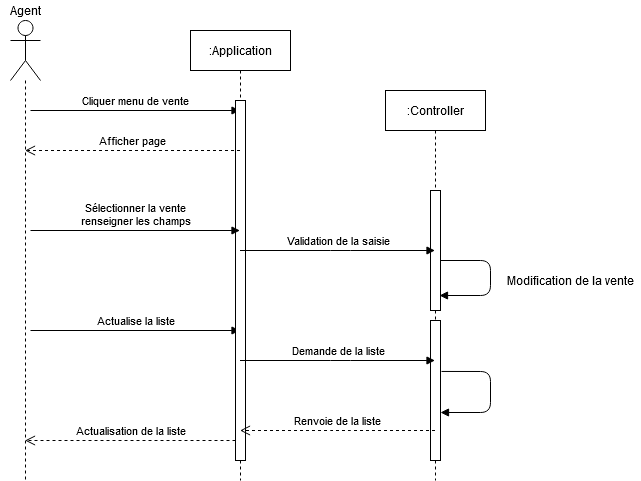
### Suppression des images d’un bien



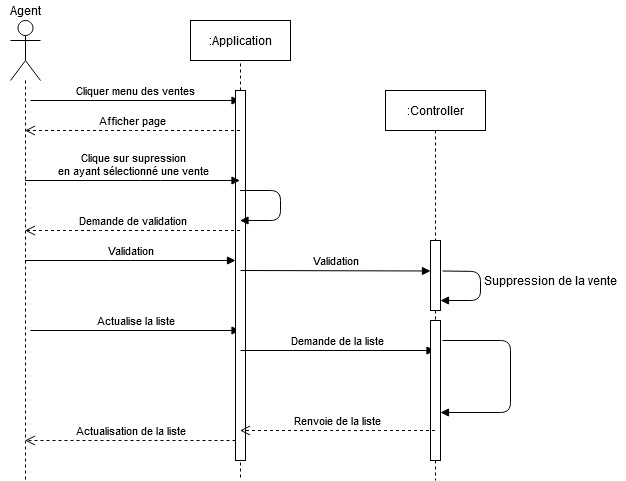
### Création d’une vente



### Modification d’une vente

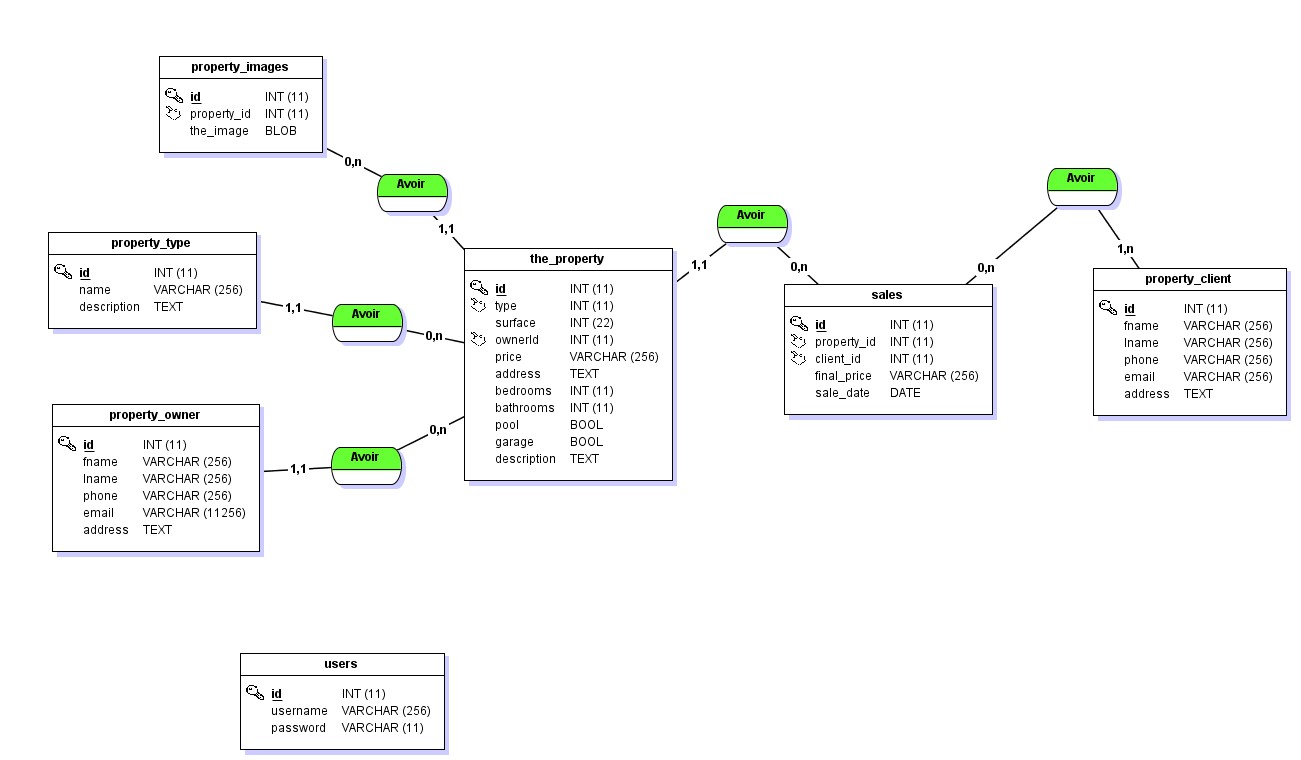


### Suppression d’une vente



## 2.4 Description des données

### 2.4.1 MCD



Le MCD représente les relations entre les différentes tables de la base de données. Cela nous permet de mieux comprendre son fonctionnement et ses interactions.

### 2.4.2 Dictionnaire de données

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Signification | Type | Taille |
| id | Id de la base de données | Int (auto increment) | 11 |
| type | Type de bien | Int | 11 |
| surface | Surface du bien | Int | 22 |
| ownerId | Id du propriétaire | Int | 11 |
| price | Prix du bien | Varchar | 256 |
| address | Adresse du bien | Varchar | 256 |
| bedrooms | Nombre de chambres | Int | 11 |
| bathrooms | Nombre de salles d’eau | Int | 11 |
| pool | Piscine présente | Boolean |  |
| garage | Garage présent | Boolean |  |
| description | Description | Text |  |
| fname | Prénom | Varchar | 256 |
| lname | Nom de famille | Varchar | 256 |
| phone | Numéro de téléphone | Varchar | 256 |
| email | Adresse email | Varchar | 256 |
| the\_image | Image stockée dans la BDD | Blob |  |
| final\_price | Prix final lors de la vente | Varchar | 256 |
| sale\_date | Date de la vente | Date |  |
| username | Nom de l’utilisateur qui se connecte | Varchar | 256 |
| password | Mot de passe de l’utilisateur qui se connecte | Varchar | 256 |

# Choix technologiques

## 3.1 Langage

Pour la réalisation de ce projet, notre choix c’est porté sur le langage Java car il s’agissait de l’une des contraintes de ce projet.

## 3.2 Logiciels utilisés

Pour développer cette application, nous avons utilisé plusieurs outils pour répondre à des besoins variés.

NetBeans : logiciel permettant de développer le code Java du projet

XAMPP : Ensemble d’outils permettant la mise en place d’une base de données rapide

GitHub : Espace de partage du code et de versioning

DrawIo : Logiciel web de création de diagrammes UML

JMerise : Logiciel de création de MCD

## 3.3 Règles de codage et convention de nommage

Nous avons pour ce projet essayer de respecter au maximum les règles historiquement proposé par Sun l’entreprise créatrice de Java.

Quelques exemples :

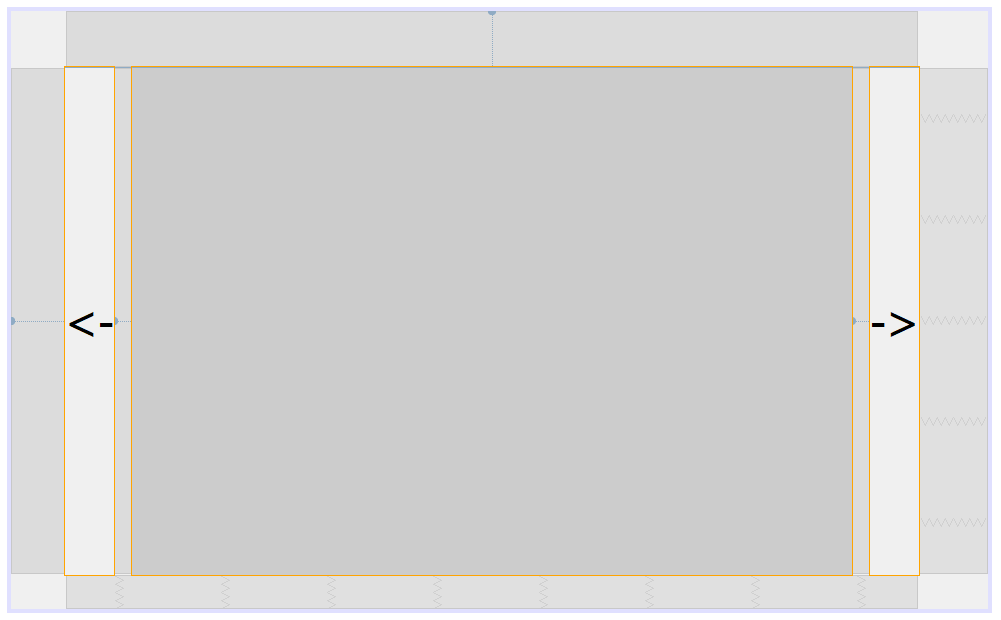
 Les méthodes contiennent un verbe « fill », la première lettre est une minuscule, chaque mot après le premier est représenté avec un majuscule et n’est pas séparé des autres mots avec un « \_ ».

 Les méthodes pour obtenir la valeur d'un champ doivent commencer par get suivi du nom du champ.

 Les méthodes pour mettre à jour la valeur d'un champ doivent commencer par set suivi du nom du champ.

# Design de l’application

Pour le design de l’application, celui-ci a été réalisé directement dans l’IDEE NetBeans qui intègre une partie gestion du design.



# Liens utiles

## 5.1 Liens des logiciels utilisées

<https://netbeans.org/>

<https://www.apachefriends.org/index.html>

<https://github.com/>

<https://draw.io>

<http://www.jfreesoft.com/JMerise/>

<https://www.jmdoudoux.fr/java/dej/chap-normes-dev.htm>

## 5.2 Lien vers le GitHub du projet

<https://github.com/Battalex/StephiPlaceSoftwareB2A>