|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Nom de naissance* |  | Romatet |
| *Nom d’usage* |  | *Entrez votre nom d’usage ici.* |
| *Prénom* |  | Mathieu |
| *Adresse* |  | 53 Grande rue saint michel 31400 Toulouse |
|  |  |

|  |  |
| --- | --- |
| **Titre professionnel visé** | |
|  | |
| Concepteur Developpeur d’Applications | |
|  | |
| **Modalité d’accès :** | |
|  | |
|  | Parcours de formation |
|  | Validation des Acquis de l’Expérience (VAE) |
|  | |

|  |  |
| --- | --- |
| **Présentation du dossier** | |
|  | |
|  | |
| Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l’emploi.**  Le DP appartient au candidat. Il le conserve, l’actualise durant son parcours et le présente **obligatoirement à chaque session d’examen**.  Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.  Il est consulté par le jury au moment de la session d’examen.  **Pour prendre sa décision, le jury dispose :**   1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l’entretien professionnel ou de l’entretien technique ou du questionnement à partir de productions. 2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle. 3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d’un parcours de formation 4. de l’entretien final (dans le cadre de la session titre).   *[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels*  *du ministère chargé de l’Emploi]*  **Ce dossier comporte :**   * pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ; * un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d’un titre, d’un diplôme, d’un certificat de qualification professionnelle (CQP) ou des attestations de formation ; * une déclaration sur l’honneur à compléter et à signer ; * des documents illustrant la pratique professionnelle du candidat (facultatif) * des annexes, si nécessaire. | |
| *Pour compléter ce dossier, le candidat dispose d’un site web en accès libre sur le site.* | |
|  | [**http://travail-emploi.gouv.fr/titres-professionnels**](http://travail-emploi.gouv.fr/titres-professionnels) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sommaire** | | | |
| **Exemples de pratique professionnelle** | | |
| **Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité** | | **p.** | **5** |
|  |  Maquette d’un gestionnaire de mails  p. | p. |  |
|  |  Intitulé de l’exemple n° 2  p. | p. |  |
|  |  Intitulé de l’exemple n° 3  p | p. |  |
|  |  |  |  |
| **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité** | | **p.** |  |
|  |  Intitulé de l’exemple n° 1  p. | p. |  |
|  |  Intitulé de l’exemple n° 2  p. | p. |  |
|  |  Intitulé de l’exemple n° 3  p | p. |  |
|  |  |  |  |
| **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | **p.** |  |
|  |  Conception d’une base de données InfoPlus (entreprise fictive) p. | p. |  |
|  |  |
|  |  |
| **Titres, diplômes, CQP, attestations de formation** *(facultatif)* | | **p.** |  |
| **Déclaration sur l’honneur** | | **p.** |  |
| **Documents illustrant la pratique professionnelle** *(facultatif)* | | **p.** |  |
| **Annexes** *(Si le RC le prévoit)* | | **p.** |  |

**Exemples de pratique**

**professionnelle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **1** | **Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité** | | | |
| ***Exemple n°1***  | | **TODO\_WebAPP** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
| Dans le cadre de la formation, il nous a été demandé de développer une application web Java EE pouvant fournir aux utilisateurs un service de gestion de liste de taches à effectuer (Todo). L’utilisateur doit pouvoir se connecter et se déconnecter à son compte via un site hébergé sur un serveur. Il pourra créer une liste de Todo, les consulter, les modifier et les supprimer. Il peut aussi faire une recherche par mot clés pour afficher les Todo. Les données doivent être persistantes.  Ce projet devait être réalisée en groupe et nous avons décidé de fonctionner en méthode Agile Scrum, la formatrice tenant le rôle de Product Owner (PO). Ça nous a permis de segmenter le projet en plusieurs étapes incrémentielles, classés par ordre d'importance, incluant un feedback à intervalles réguliers et une mise à jour des tickets JIRA. Cette méthode a permis de favoriser la collaboration, de procéder à des ajustements à intervalles réguliers afin de s'assurer de répondre aux besoins. Nous avons aussi créé un dépôt distant sur GitLab pour le contrôle de version.  **Première étape : conception**  Nous avons dans un premier temps utilisé le langage de représentation UML destiné à la modélisation objet. | | | | | |
| Modèles fonctionnels  **Diagramme de cas d’utilisation** (ou Use Cases) qui décrit les besoins d’utilisation du système, en privilégiant le point de vue de l’utilisateur (ex : Annexe 1)  **Diagramme d'activité** qui montre l'enchaînement des activités qui concourent au processus (ex: Annexe 2)  Modèles statiques  **Diagramme de classes** qui décrit la structure des classes et les relations qui existent entre elles (ex: Annexe 3)  Modèles dynamiques  **Diagramme de séquence** qui décrit la séquence par scénario de messages échangés entre des objets (ex: Annexe 4)  Modèles architecturaux  **Diagramme de déploiement** qui montre la configuration physique des matériels du système (ex : Annexe 5)  Une fois la modélisation faite, nous avons réalisé des maquettes IHM (Annexe 6) et les avons fait valider par le PO.  **Deuxième Etape : Développement**  Pour ce projet, le design pattern **Modèle / Vue / Contrôleur** (MVC) sera appliqué. Le modèle **MVC** offre une séparation claire en couche des responsabilités au sein d’une application, en conformité avec les principes de conception : responsabilité unique, couplage faible et cohésion forte   * **La partie Modèle** (Java Bean) correspond à la logique métier. Elle représente le comportement de l’application : traitements des données, interactions avec la base de données, etc. elle assure la gestion de ces données et garantit leur intégrité. * **La partie Vue** (JSP) correspond à l’interface graphique avec laquelle l’utilisateur interagit. La vue a pour rôle d’afficher et de présenter les données ou les résultats renvoyés par le modèle. Elle n’effectue aucun traitement et est une page écrite en HTML et JAVA. * **La partie Contrôleur** (Servlet) gère les requêtes HTTP, demande au Modèle de trouver les données correspondantes dans la base de données, analyse les données fournies par le Modèle et décide ce qui doit être affiché par la partie Vue.     DAO  Maven est un outil open source de la communauté Apache entièrement écrit en Java. Il permet d’automatiser la gestion et la construction d’un projet Jav | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| Diagramme de cas d’utilisation : visual-paradigm.com | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Cliquez ici pour taper du texte.* | |
| Chantier, atelier, service | | | *Cliquez ici pour taper du texte.* | | |
| Période d’exerciceDu : *Cliquez ici*au :*Cliquez ici* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité** | | | |
| ***Exemple n° 1***  | | ***Projet InfoPlus*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Le projet InfoPlus consistait à concevoir et développer une Base de Données relationnelle pour la gestion de stock de la société « InfoPlus ». On m’a fourni un cahier des charges contenant l’expression du besoin et on pouvait interroger le Maitre d’ouvrage si besoin. Pour cela j’ai utilisé la méthode **MERISE** pour arriver à concevoir un Système d'Information (**SI**) en séparant les données et les traitements à effectuer en plusieurs modèles conceptuels, logique et physique.   **Première étape : Analyse des données (annexe 1)**  Cette étape consiste à analyser toutes les informations du cahier des charges et interroger le maitre d’ouvrage si besoin de précision dans le but d’établir :   * Le **dictionnaire des données** qui est un tableau qui fait l'inventaire des éléments nécessaires au SI et leurs caractéristiques (entités, identifiant, nom, type, taille, persistés ou calculés, règles de calcul, etc.) * **Les règles de Gestion** qui à l’échelle de l’entreprise va s’appliquer systématiquement dans les cas qu’elle doit régir.   **Deuxième étape : Modèle Conceptuel des Données (annexe 2)**  J’ai ensuite élaboré le MCD permettant de décrire le SI à l'aide d'entités et d’associations en contrôlant les règles de normalisation (nom **entité**, **association** et **attribut** uniques, présence d’identifiant, **cardinalité** des associations)  **Troisième étape : Modèle Logique des Données (annexe 2)**  A partir du MCD, j’ai fait le MLD qui est la représentation de la structure de la base de données (transformation en **tables**, **champs** et **clés primaires**, migration des **clés étrangères**.  **Quatrième étape : Modèle Physique des Données**  Elaborer le MPD consiste à implémenter le modèle dans le **SGBD** (PostgreSQL). Pour cela j’ai créé les scripts suivants que j’ai exécuté en m’aidant de l’outil **pgAdmin**.  1) Créer un utilisateur et la base de données avec **Postgres psql** : | | | | | |
| 2) Script de création des tables et contraintes sur pgAmin :  Ici on peut voir la création des tables T\_INDIVIDU et T\_EMPLOYE en spécifiant leurs types, longueurs, clés primaires et contraintes.    3) Script d’insertion des données, ici un extrait pour la table T\_EMPLOYE :    4) Puis j’ai élaboré des **procédures ou fonctions stockées**, **triggers** et **vues** pour répondre aux besoins exprimés :    Exemple Fonction Stockée :    Exemple trigger :  Exemple Vue : | | | | | |
|  | | | | |
| 5) Exécution des scripts :  Afin d’assurer le bon fonctionnement de l’ensemble, la phase de test a été effectuée à l’aide de scripts manuels exécutés dans un ordre précis :  01-Script\_CreationUtilisateur.sql  02-Script\_SuppressionTables.sql  03-Script\_CreationTables.sql  04-Script\_CreationFonctionsProcédures.sql  05-Script\_CreationTriggers.sql  06-Script\_CreationVues.sql  07-Script\_InsertionDonnees.sql | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| Conception de la Base de Données :   * Dictionnaire des données sous Excel * MCD et MLD sous Looping   Mise en place de la BDD :   * PostgresSQL avec pgAdmin 4   Développement des composants :   * Langage PL/PGSQL | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Cliquez ici pour taper du texte.* | |
| Chantier, atelier, service | | | *Cliquez ici pour taper du texte.* | | |
| Période d’exerciceDu : *Cliquez ici*au :*Cliquez ici* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **3** | **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** | | | |
| ***Exemple n° 1***  | | ***E-commerce*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
| Pour notre back-end Spring Boot, nous avons tiré parti de Spring Data REST pour minimiser le code et créer une API REST. Donc, Spring Data REST analysera essentiellement notre projet  pour un JpaRepository,  et il exposera les API REST pour chaque type d'entité  pour notre JpaRepository.  Voici donc un exemple ici.  Nous avons ce ProductRepository étend JpaRepository.  Nous donnons au type d'entité virgule le type de clé primaire.  Le type d'entité ici est donc Produit.  Le type de clé primaire est Long.  Et sur la base de ces informations,  il exposera les points de terminaison REST.  Donc, un peu plus ici sur les points de terminaison REST.  Par défaut, Spring Data REST créera des points de terminaison  en fonction du type d'entité.  Il utilisera le formulaire simple et pluralisé.  Il faudra donc essentiellement le premier caractère  du type d'entité et le rendre en minuscules  puis ajoutez simplement un S à l'entité.  Donc, dans notre exemple de notre référentiel de produits  étend la virgule du produit JpaRepository Long.  Cela exposera essentiellement un point de terminaison appelé  slash produits.  Nous allons donc prendre le produit, ajoutez-y simplement un S.  Donc, ce qui est vraiment bien ici, c'est que Spring Data REST  exposera ces points de terminaison gratuitement.  Nous aurons donc essentiellement une barre oblique produits,  et nous aurons un support pour tous les différents  Méthodes HTTP pour POST, GET, PUT, DELETEs.  Donc, cela nous donnera tout notre soutien CRUD  pour nos entités données,  pour créer, lire, mettre à jour et supprimer.  Donc, très peu de code ici pour exposer votre API REST  en utilisant Spring Data REST et aussi avec Spring Boot. | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *Cliquez ici pour taper du texte.* | |
| Chantier, atelier, service | | | *Cliquez ici pour taper du texte.* | | |
| Période d’exerciceDu : *Cliquez ici*au :*Cliquez ici* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |
| --- | --- | --- |
| **Titres, diplômes, CQP, attestations de formation** | | |
|  | | |
| *(facultatif)* | | |
| **Intitulé** | **Autorité ou organisme** | **Date** |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Déclaration sur l’honneur** |
|  |
|  |

Cliquez ici pour taper du texte.

Je soussigné(e) [prénom et nom] ,

déclare sur l’honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l’auteur(e) des réalisations jointes.

Cliquez ici pour taper du texte.

Cliquez ici pour choisir une date

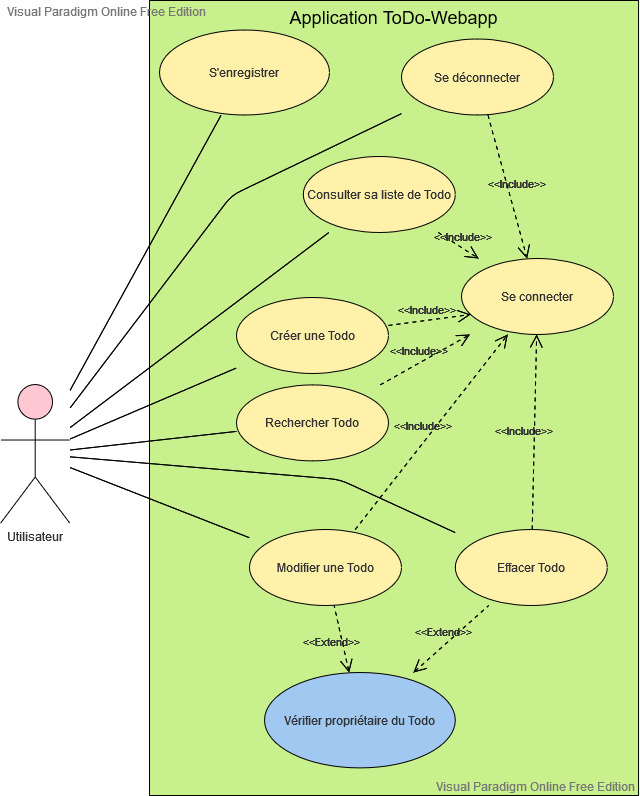
Fait à le

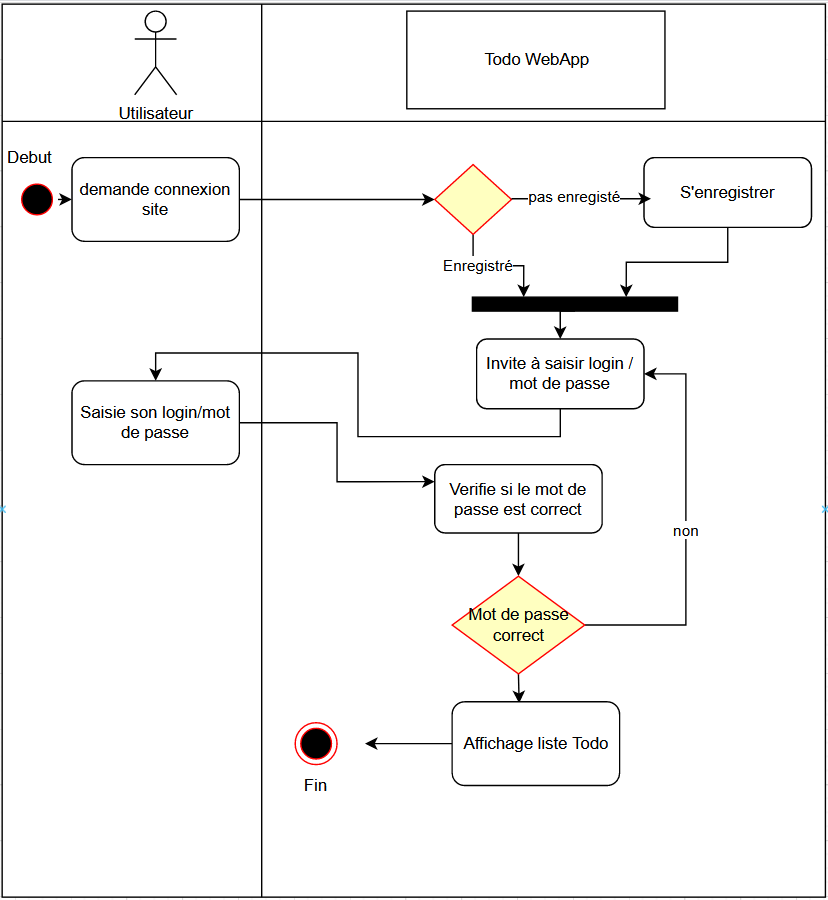
pour faire valoir ce que de droit.

Signature :

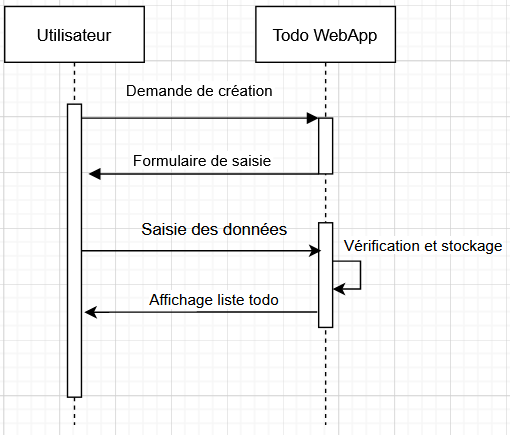
|  |
| --- |
| **Documents illustrant la pratique professionnelle** |
|  |
| *(facultatif)* |
| **Intitulé** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

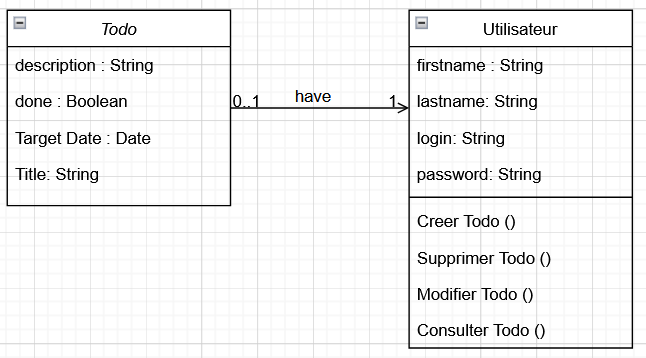
|  |
| --- |
| **Annexes** |
|  |

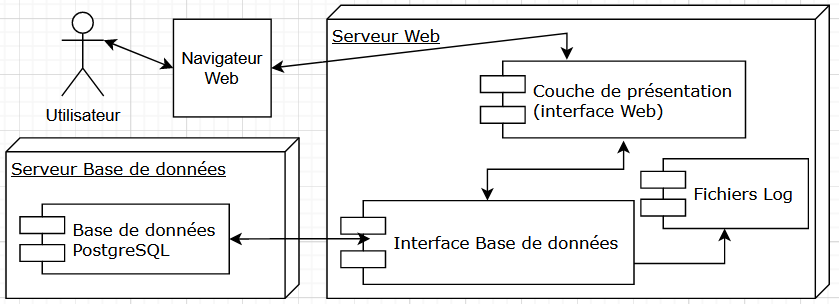
*Annexe 1 : Diagramme des Cas d’utilisation (use case) : Todo WebApp*

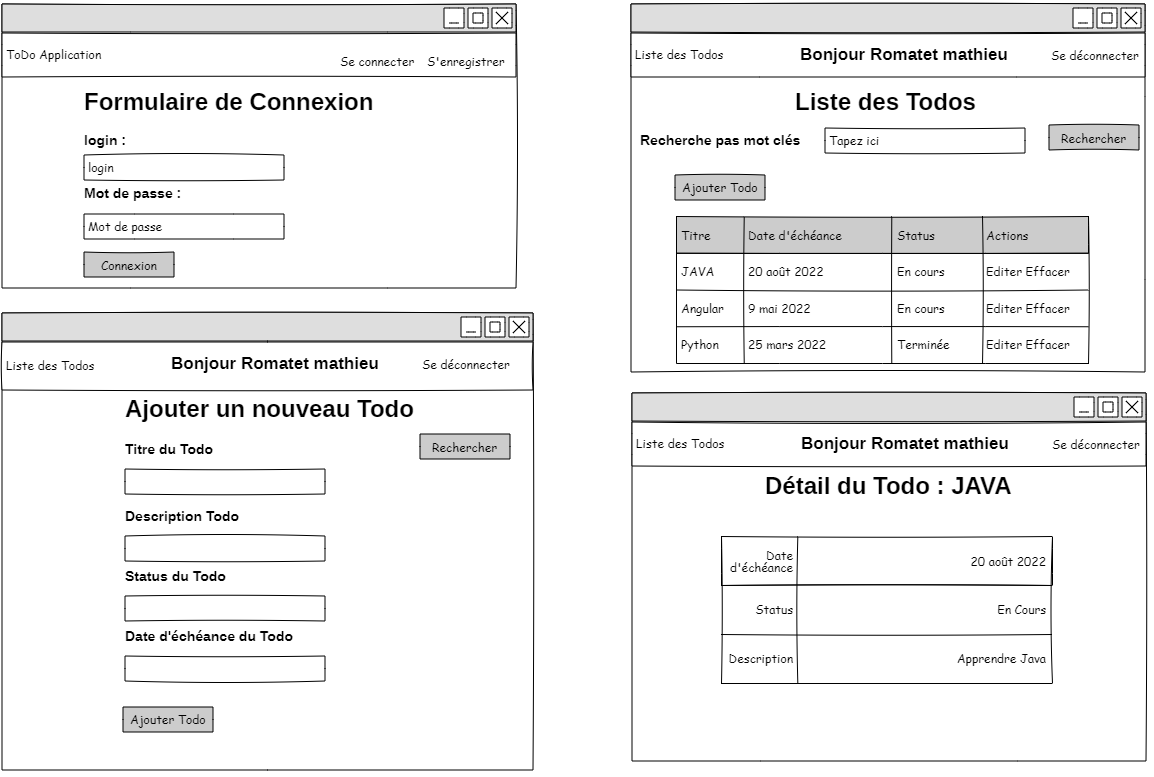


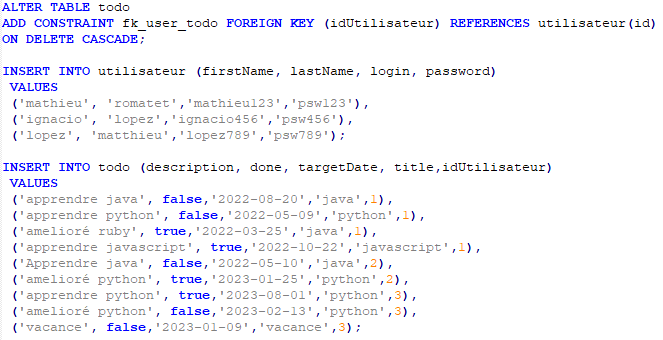
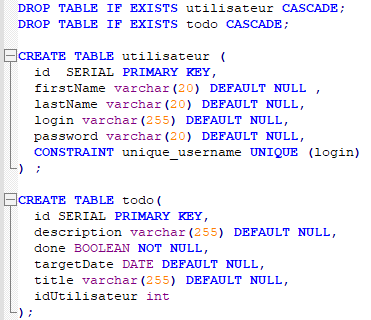
*Annexe 2 : Diagramme d’Activité : Connexion* ***Todo WebApp***

*Annexe 3 : Diagramme de séquence : création d’une Todo :* ***Todo WebApp***

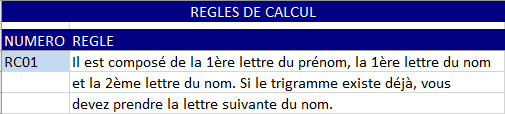
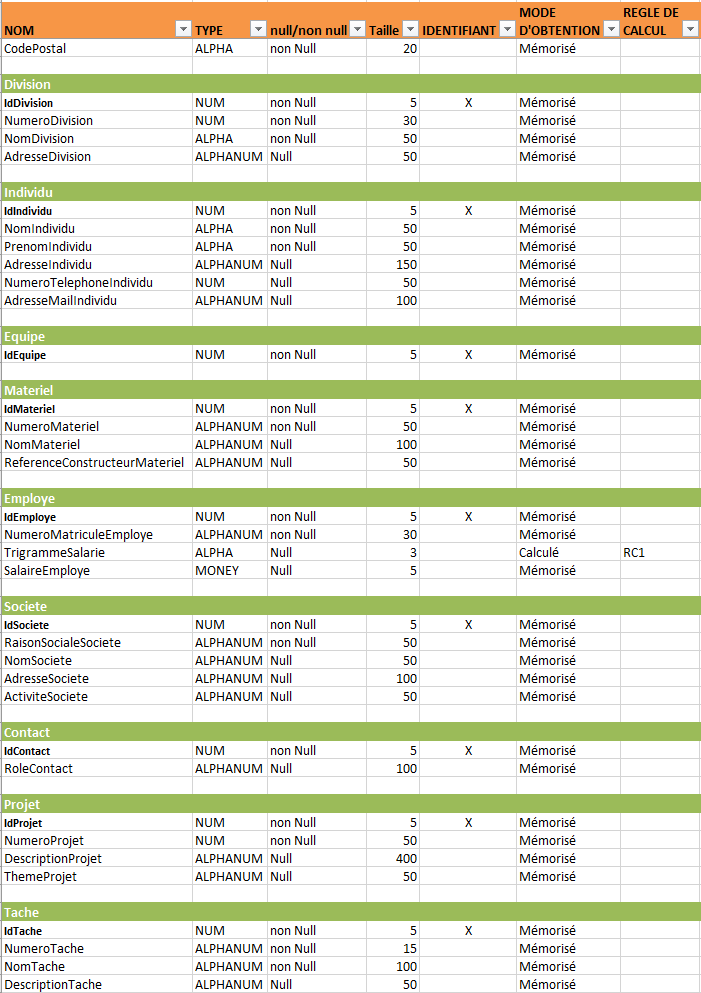
*Annexe 4 : Diagramme de Classe :* ***Todo WebApp***

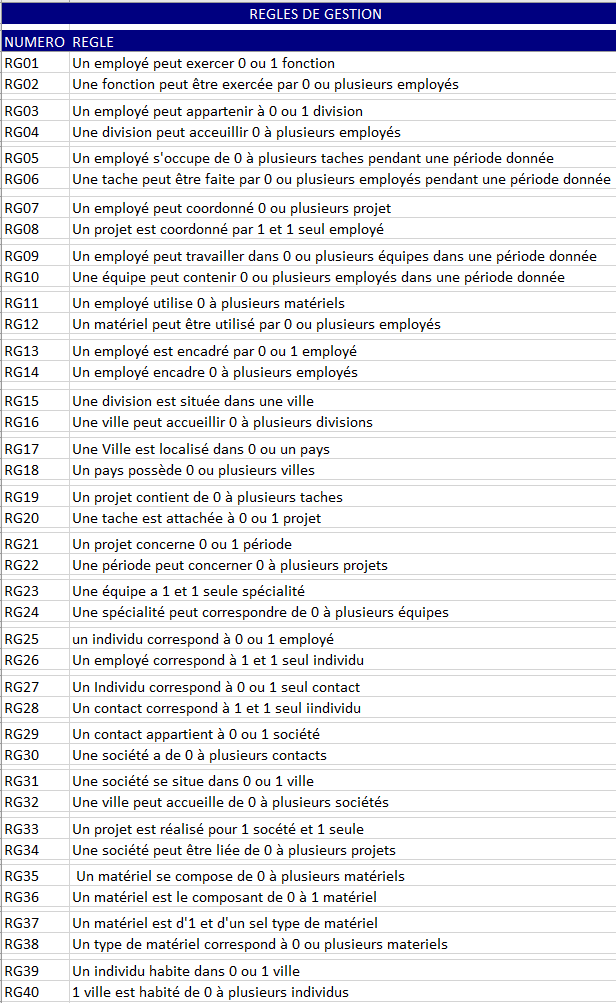
*Annexe 5 : Diagramme de déploiement :* ***Todo WebApp***

*Annexe 6 : Maquettes* ***Todo WebApp***

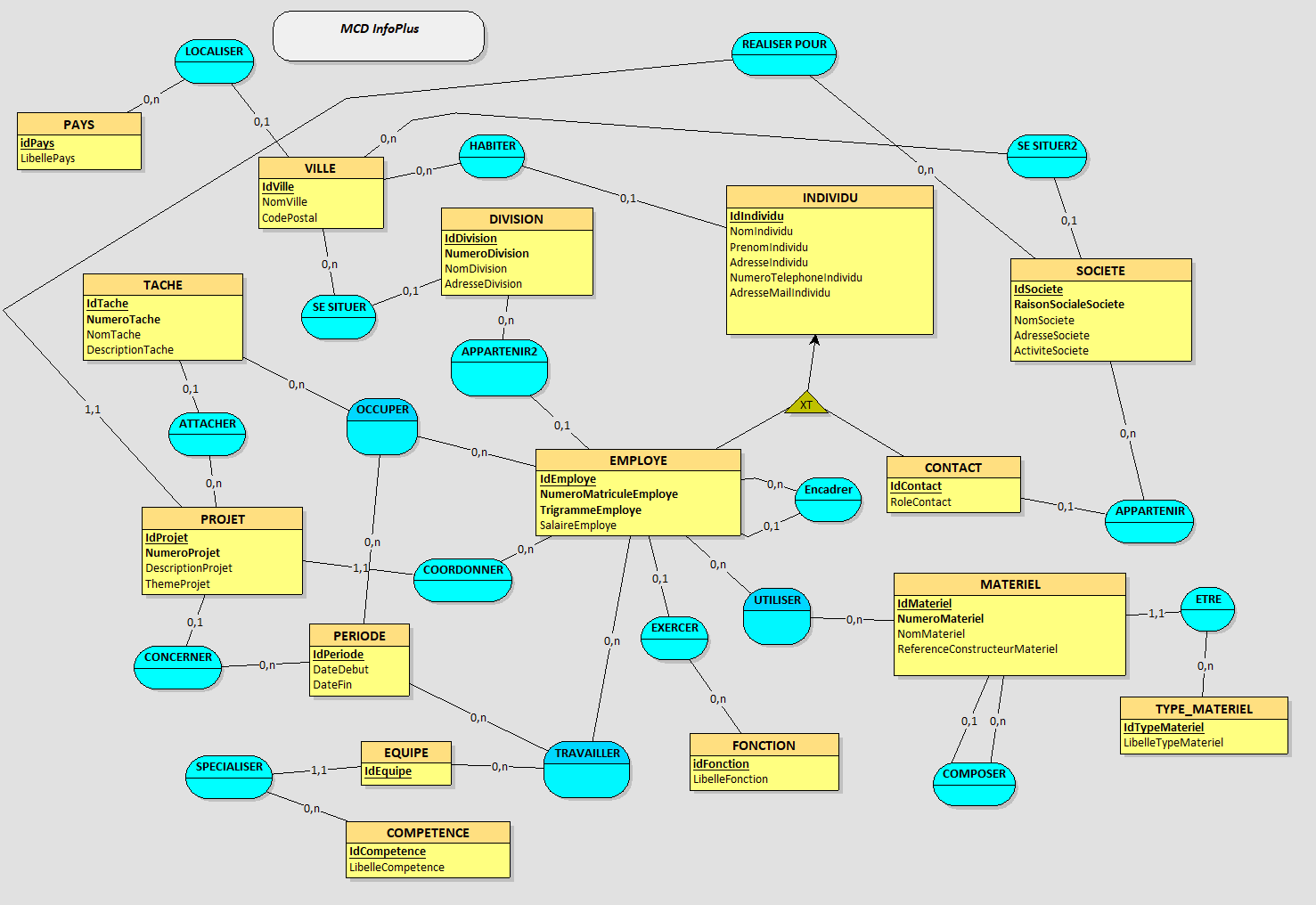
*Annexe 7 : Script création Base de données* ***Todo WebApp***

*Annexe  : Dictionnaire des données* ***InfoPlus***

****



*Annexe 2 : MCD et MLD de la base de données* ***InfoPlus***

**

