



---

# SKILLZ

---

Application de validation et de suivi des compétences



**Réalisé par :**

Bérénice GNAMBLOHOU  
Riad LAZLI  
Dung LE THUY

**Encadré par :**

Vincent GROLLEMUND  
Pascal POIZAT  
Sonia SAADAOU

## Table des matières

Introduction .....	2
1. Contexte et objet du projet .....	3
2. Détail fonctionnel .....	3
3. Détail Technique.....	11
A. Cas d'usage .....	11
B. Conception Architecturale générale du logiciel.....	12

## Introduction

Dans le cadre du module projet SI et Données, nous avons décidé de créer SKILLZ, une application de validation et de suivi des compétences.

L'objectif de ce module est de mener un projet informatique depuis la conception jusqu'au déploiement en suivant une méthodologie Agile et Dev OPS.

Il nous permettra d'une part de mettre en application toutes les connaissances acquises au cours de cette formation (management de projet, programmation et intelligence des données), et d'autre part, acquérir de nouvelles connaissances par l'utilisation de nouvelles technologies.

Ce document représente le cahier des charges de l'application SKILLZ, imaginé par notre équipe. Nous y détaillerons le contexte du projet, les types d'utilisateurs et les différentes fonctionnalités attendues dans chaque itération en présentant notre vision de la solution ainsi que les contraintes que nous devons respecter.

## 1. Contexte et objet du projet

Le projet SKILLZ consiste à développer une application web de validation et de suivi des compétences des employés au sein d'une entreprise ou des étudiants au sein d'une université.

En effet, il est difficile pour un évaluateur (manager, recruteur, commercial, enseignant ...) d'évaluer de façon concrète le niveau de maîtrise pour une compétence donnée d'un candidat qu'il reçoit, surtout si ce dernier n'a aucune preuve à lui fournir, cela peut être par exemple un diplôme officiel, une certification de réussite à une formation...

Pour répondre à cette problématique, l'application SKILLZ permettra de s'assurer des compétences d'un candidat à travers la création de questionnaires personnalisés. Elle pourra être utilisée aussi bien dans un environnement d'entreprise (entreprise de prestations ou cabinets de conseils) en permettant au manager de s'assurer et de suivre les compétences des collaborateurs, que dans l'éducation en permettant ainsi aux enseignants de s'assurer de la bonne compréhension des cours et de suivre l'évolution des compétences des étudiants.

## 2. Détail fonctionnel

On identifie 3 types d'utilisateurs, à savoir :

- **Admin :**

L'utilisateur de type Admin est une personne qui pourra administrer l'application avec des droits d'accessibilité à des sections spécifiques. Il aura pour but, la gestion de l'application.

Pour éviter les inscriptions, l'Administrateur pourra créer des utilisateurs en leur attribuant des droits (Concepteur ou Participant). Il pourra créer le référentiel des compétences.

Enfin, il aura un tableau de bord dédié, qui lui permettra d'avoir une vision globale sur l'application (liste des concepteurs, liste des compétences, liste des questionnaires...).

Dans le contexte d'une utilisation au sein d'une université, on considère que l'admin est l'enseignant ou bien le secrétariat. Dans un environnement d'entreprise, l'admin représente le Manager ou bien le responsable des ressources humaines.

SkillZ

← → ↻ 🔍 https://SkillZ.com ☰

### SkillZ - Application de gestion de compétences

Identifiant

Mot de passe

*Figure 1 : Utilisateurs - Connexion*

SkillZ

← → ↻ 🔍 https://SkillZ.com ☰

[Deconnecter](#)

Nombre d'utilisateurs : 100

Nombre de concepteurs : 10

Nombre de participants : 90

Nombre de compétences : 30

[L'arbre de compétences](#)

*Figure 2 : Admin-Accueil*

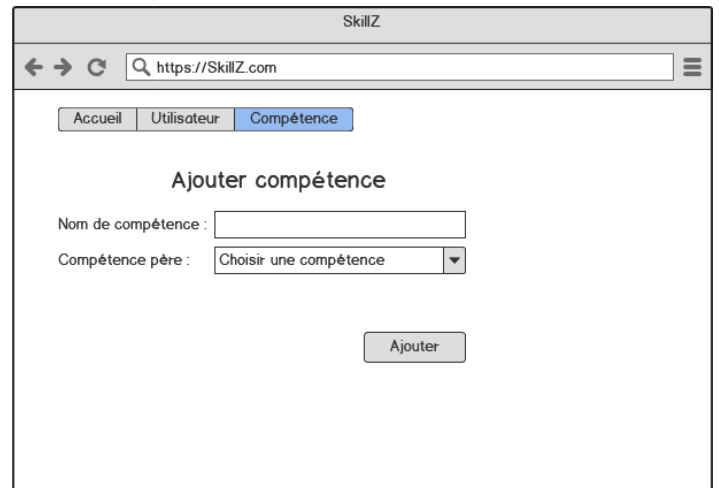


Figure 3 : Admin- Ajout de compétences

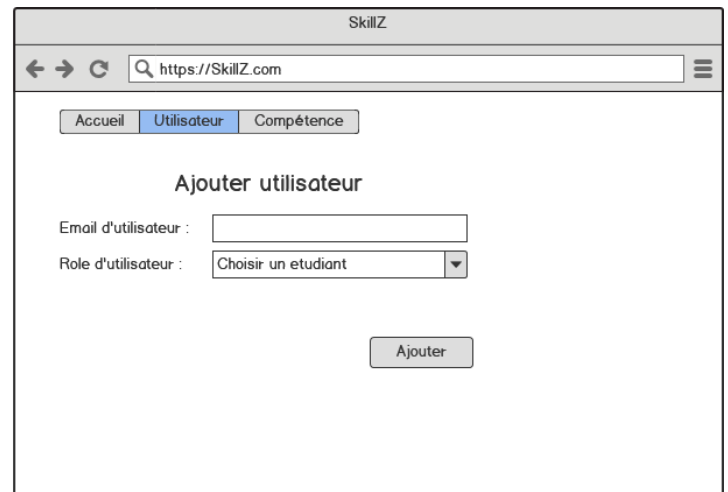
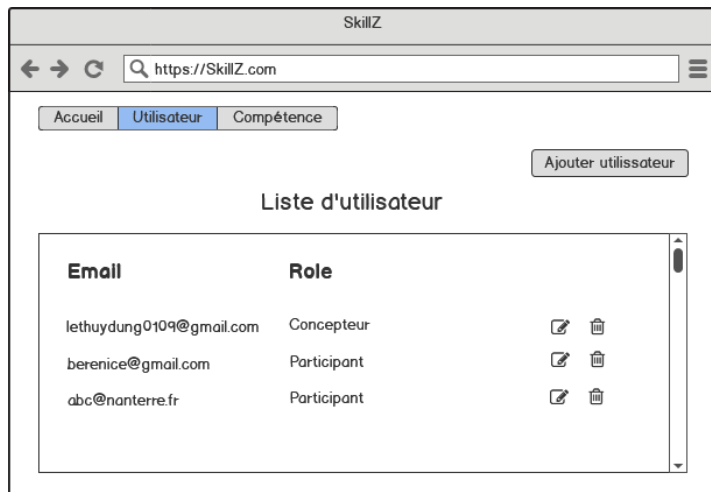


Figure 4 : Admin - Ajout utilisateur

#### - **Concepteur :**

L'utilisateur de type concepteur est soit :

Un manager si l'application est utilisée au sein d'une entreprise, ou bien un enseignant si cette dernière est utilisée dans un contexte universitaire.

Le concepteur peut créer des questions pour un domaine de compétence et un niveau de maîtrise définis (par exemple créer une question pour la programmation orientée objet pour un niveau 3 de maîtrise). Ces questions seront ajoutées à une bibliothèque de questions qui contient déjà des questions créées par d'autres concepteurs.

Sur la base la bibliothèque de questions, le concepteur pourra créer des questionnaires de validation des compétences.

Enfin, Il pourra visualiser les résultats des participants aux questionnaires et suivre l'évolution des tendances en termes de compétences.

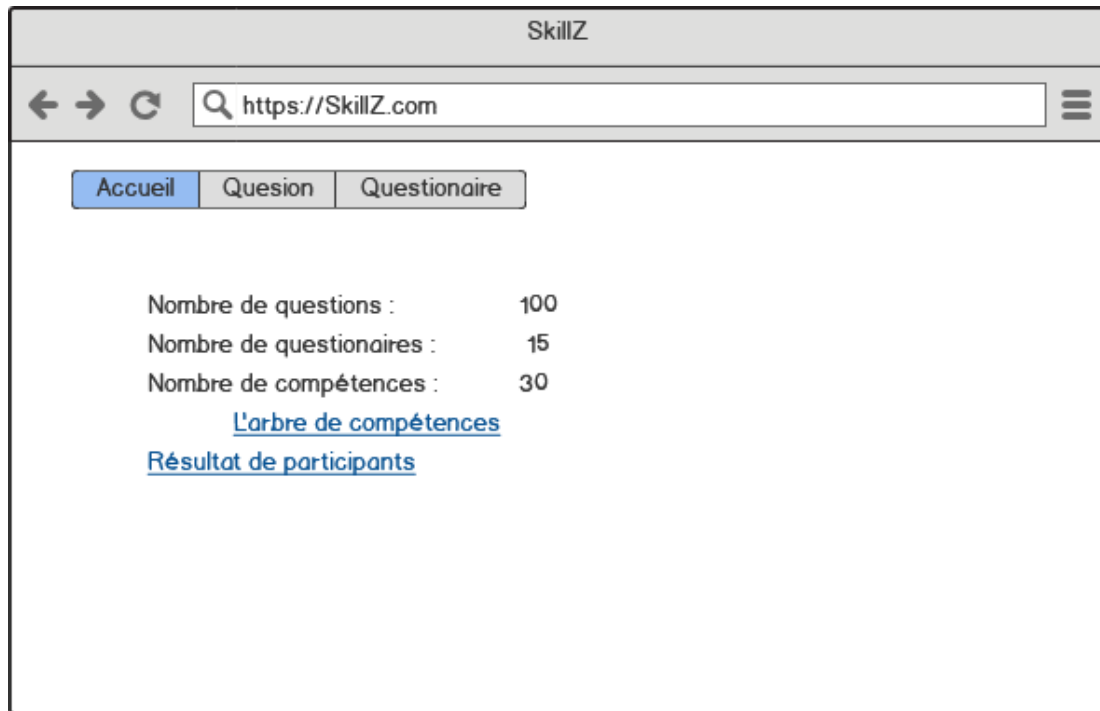


Figure 5 : Concepteur - Accueil

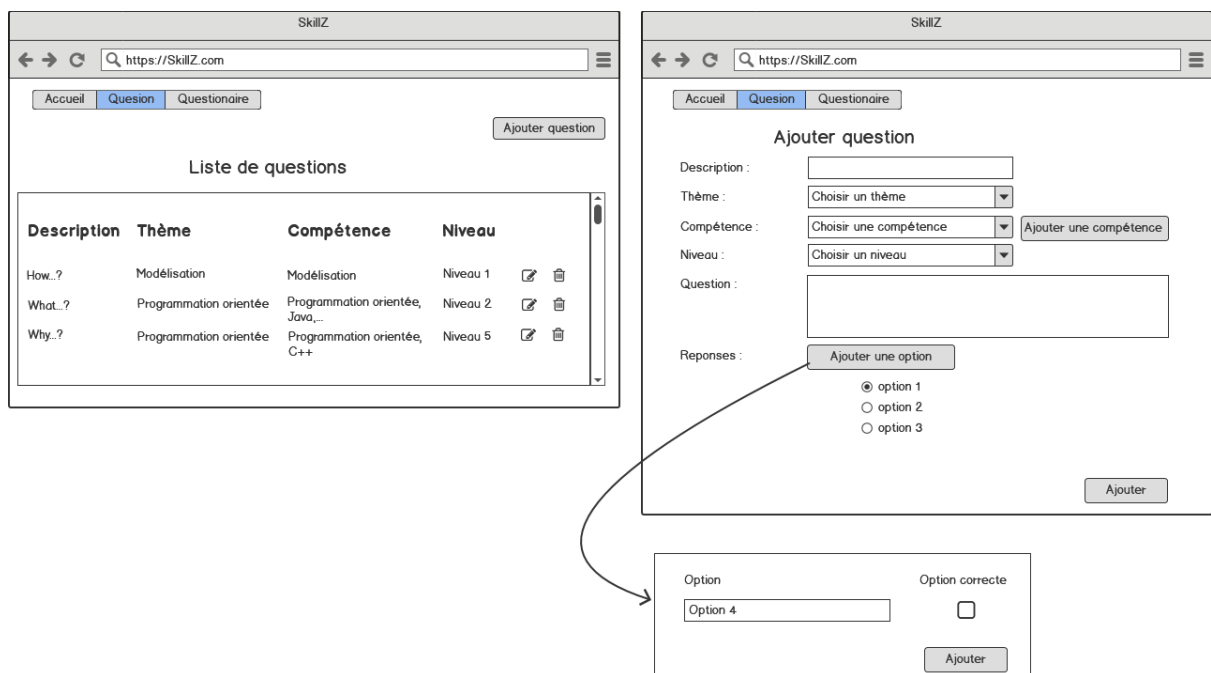


Figure 6 : Concepteur - Création d'une question

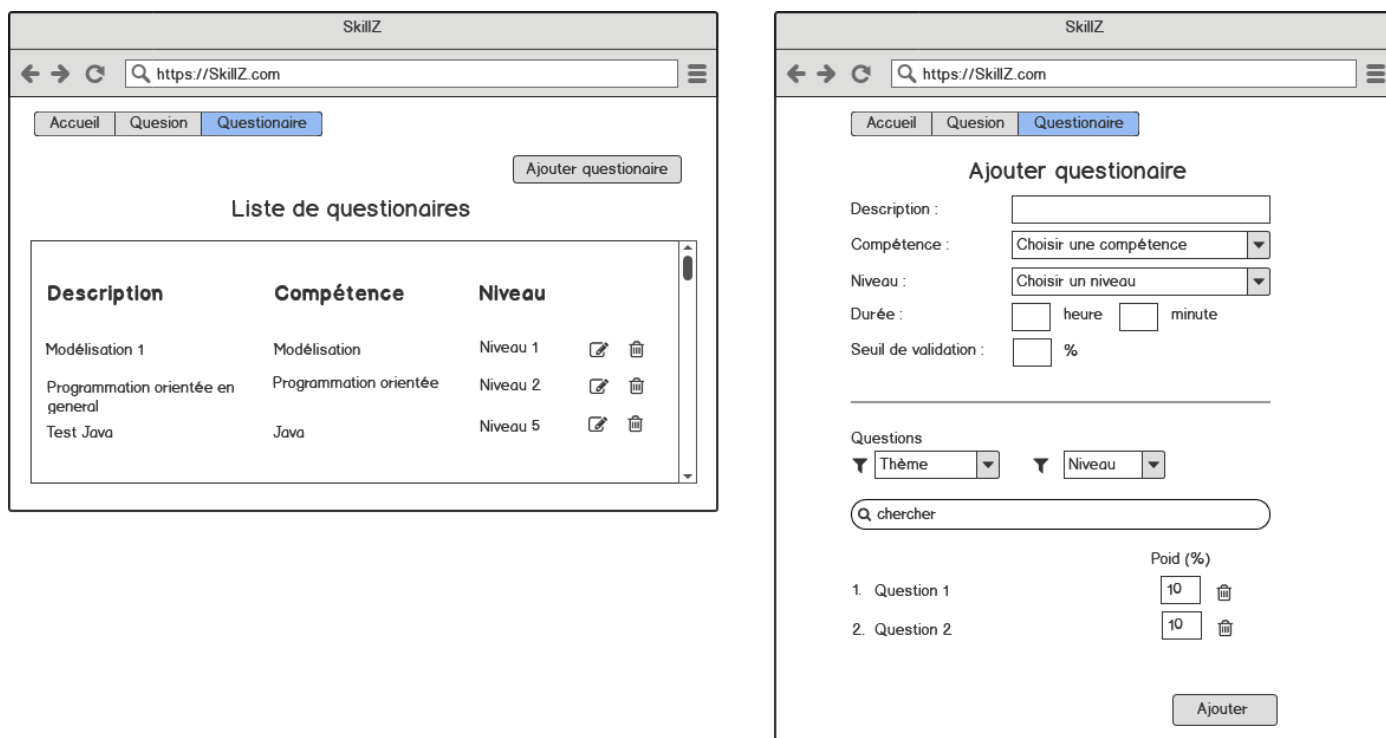


Figure 7 : Concepteur - Création d'un questionnaire

#### - Participant :

Si l'application est utilisée au sein d'une entreprise, le participant représente un collaborateur. Sinon, si elle est utilisée dans une université, le participant est un étudiant.

Le participant aura pour objectif de mettre à jour ses compétences sur son profil. En effet, il devra choisir dans la liste des compétences disponibles, puis celle à ajouter à son profil.

L'affichage d'une compétence sur le profil (avec le niveau de maîtrise) ne se fera que lorsque l'utilisateur aura passé l'un des tests de validation (créer par les concepteurs) associés à ladite compétence.

Pour chaque questionnaire, le participant aura un temps limité et un seuil minimal pour valider le questionnaire.

Quand le participant réussit un questionnaire, il obtient un nouveau badge sur son profil qui certifie son niveau de maîtrise pour la compétence évaluée à la date du jour (par exemple, le participant réussit un test de niveau 4 en UML. Il aura sur son profil le badge UML-Niveau4 avec à côté la date de réussite du questionnaire).

Dans le cas contraire, l'accès au questionnaire qu'il a raté, sera bloqué pendant 1 mois.



Par ailleurs, si le participant passe plusieurs fois un test de même niveau, la date de réussite de son test est mise à jour s'il réussit le nouveau test sinon la date de réussite reste inchangée.

De plus, si l'utilisateur passe plusieurs tests pour la même compétence mais pour des niveaux différents alors son badge est mis à jour avec le niveau le plus élevé en cas de réussite (par exemple j'ai validé ma compétence java au niveau 2. Si je valide un test java de niveau 3 alors mon badge passe de Java-Niveau 2 à Java-Niveau 3 et si j'échoue alors mon badge reste au niveau 2).

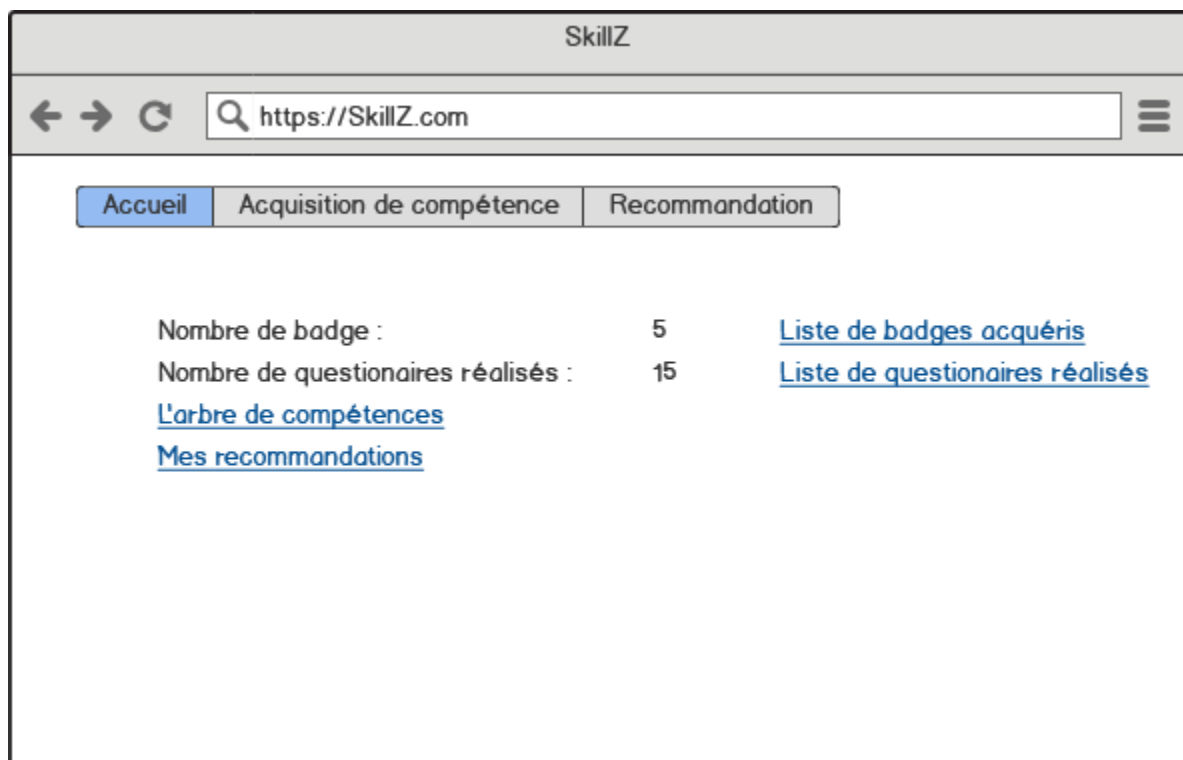


Figure 8 : Participant - Accueil

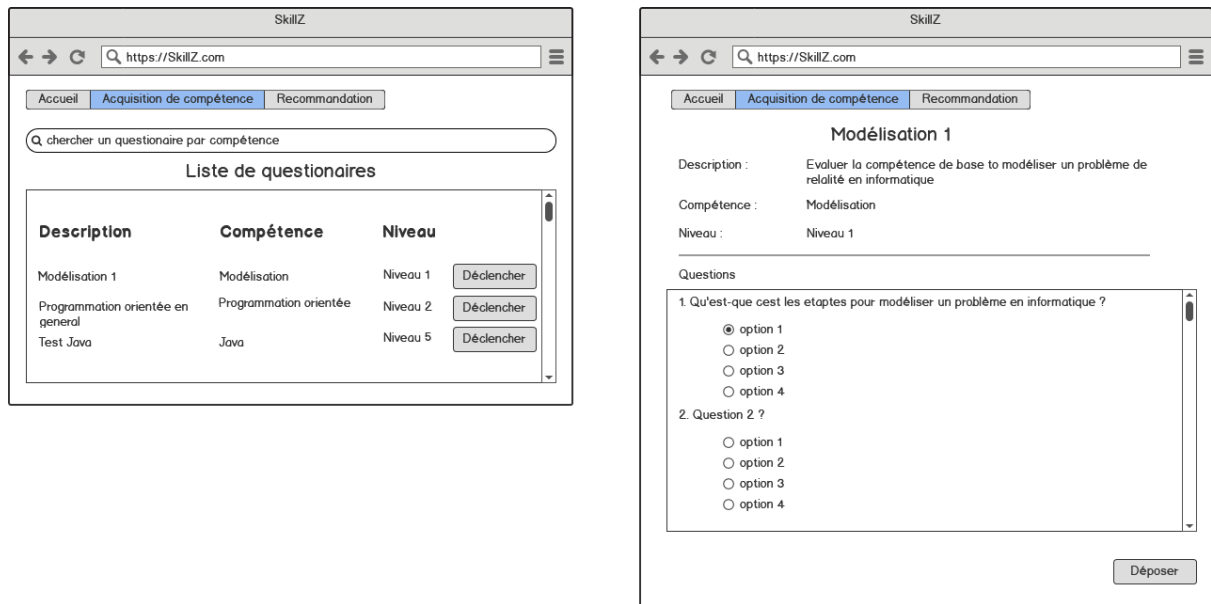


Figure 9 : Participant - Passage de questionnaire

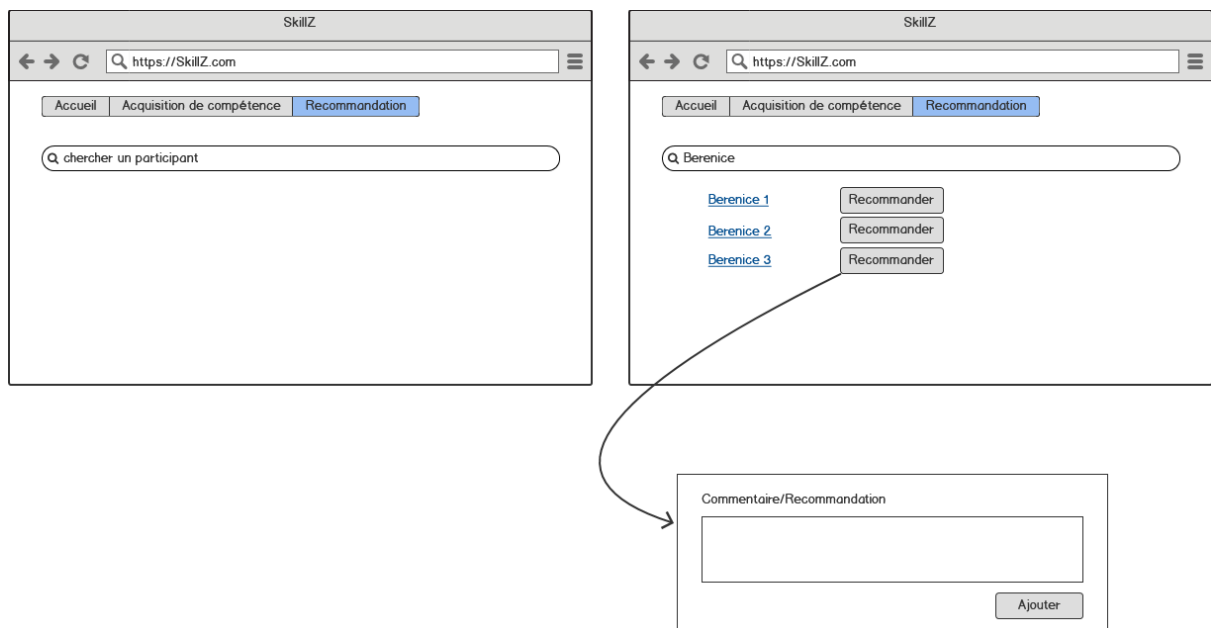


Figure 10 : Participant - Recommandation

Le tableau ci-dessous fait le récapitulatif de chaque utilisateur avec ses fonctionnalités.

Type utilisateur	Fonctionnalités
Admin	<ul style="list-style-type: none"> <li>- Ajouter/Supprimer/Modifier un/une liste de collaborateur(s)</li> <li>- Ajouter/Supprimer/Modifier un référentiel de compétences</li> </ul>
Concepteur (Manager et/ou Enseignant)	<ul style="list-style-type: none"> <li>- Créer/modifier/supprimer une/des question(s)</li> <li>- Créer/modifier/supprimer un questionnaire de validation des compétences</li> <li>- Rechercher des participants selon des filtres</li> <li>- Visualiser l'évolution des tendances en matière de compétences via un tableau de bord</li> </ul>
Participant (Collaborateur et/ou étudiant)	<ul style="list-style-type: none"> <li>- Visualiser/Modifier les informations personnelles (adresse, numéro de téléphone...) sur le profil</li> <li>- Ajouter/Supprimer/ une compétence de son profil</li> <li>- Passer un test pour valider une compétence</li> <li>- Recommander un collaborateur</li> </ul>

En nous basant sur le planning qui nous a été fourni par notre enseignant, nous avons défini dans le tableau ci-dessous les fonctionnalités à implémenter par itération.

Release	Date de rendu	Fonctionnalités
Itération 1 (v0.0) : Préparation et mise en place	12/03	<ul style="list-style-type: none"> <li>- Rédaction du cahier de charges et spécifications techniques</li> <li>- Mise en place de l'infrastructure technique du projet (Repo GitHub, outil de tests, installation des librairies.)</li> </ul>
Itération 2 (v1.0) : Mise en place du système de base	09/04	<ul style="list-style-type: none"> <li>- Créer/Modifier/Supprimer un utilisateur</li> <li>- Connexion</li> <li>- Modifier les informations du profil</li> <li>- Créer un référentiel de compétences</li> <li>- Créer/ Modifier/Supprimer une question</li> <li>- Créer/Modifier/Supprimer un questionnaire (en sélectionnant des questions disponibles dans la bibliothèque)</li> </ul>
Itération 3 (v1.1)  Mise en place du système de passage et validation des tests, attribution de badges	16/04	<ul style="list-style-type: none"> <li>- Ajouter/Supprimer une compétence sur son profil</li> <li>- Passer un test avec les règles de validations (chronomètre, seuil minimal de réussite...)</li> <li>- Attribuer un badge de compétence avec un niveau égal à celui du test</li> </ul>

		<ul style="list-style-type: none"> <li>- Rechercher un participant selon un filtre, et visualiser ses résultats</li> </ul>
Itération 4 (v1.2) Système de recommandations	30/04	<ul style="list-style-type: none"> <li>- Ajouter/Modifier/Supprimer un avis pour un participant</li> <li>- Afficher les recommandations sur le profil d'un participant</li> </ul>
Itération 5 (v1.3) : Mise en place de l'intelligence des données	03/05	<ul style="list-style-type: none"> <li>- Amélioration de l'arbre des compétences</li> <li>- Visualisation des statistiques de l'application et de l'évolution des compétences via un tableau de bord</li> </ul>
Itération 6 (v1.4) : Finalisation et clôture		<ul style="list-style-type: none"> <li>- Corriger, améliorer et finaliser les fonctionnalités en place.</li> </ul>

### 3. Détail Technique

#### A. Cas d'usage

La figure ci-dessous représente le diagramme de use case de l'application SKILLZ.

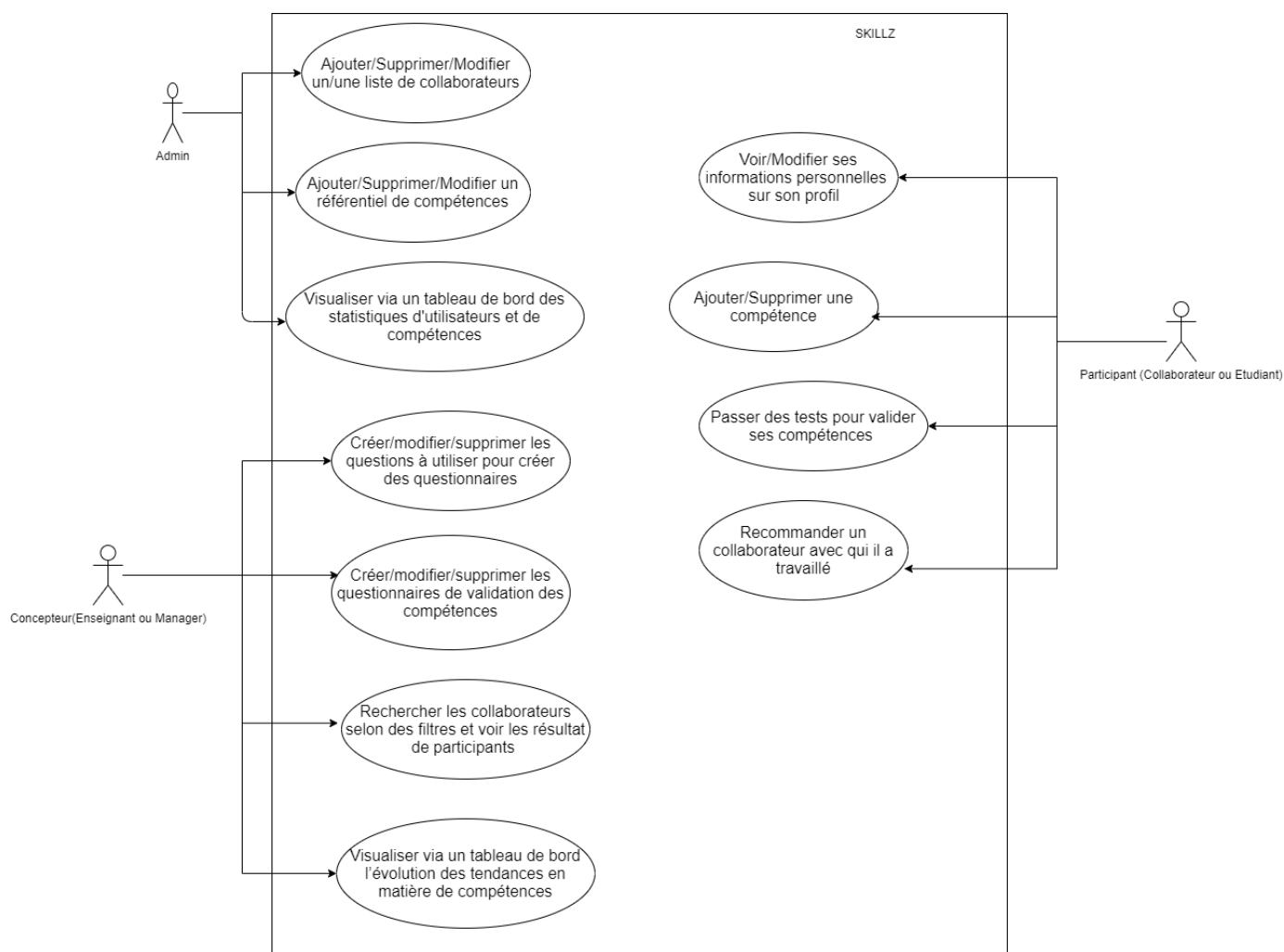


Figure 10 : Diagramme de use case

## B. Conception Architecturale générale du logiciel

Le diagramme ci-dessous représente le diagramme de classe de notre application.

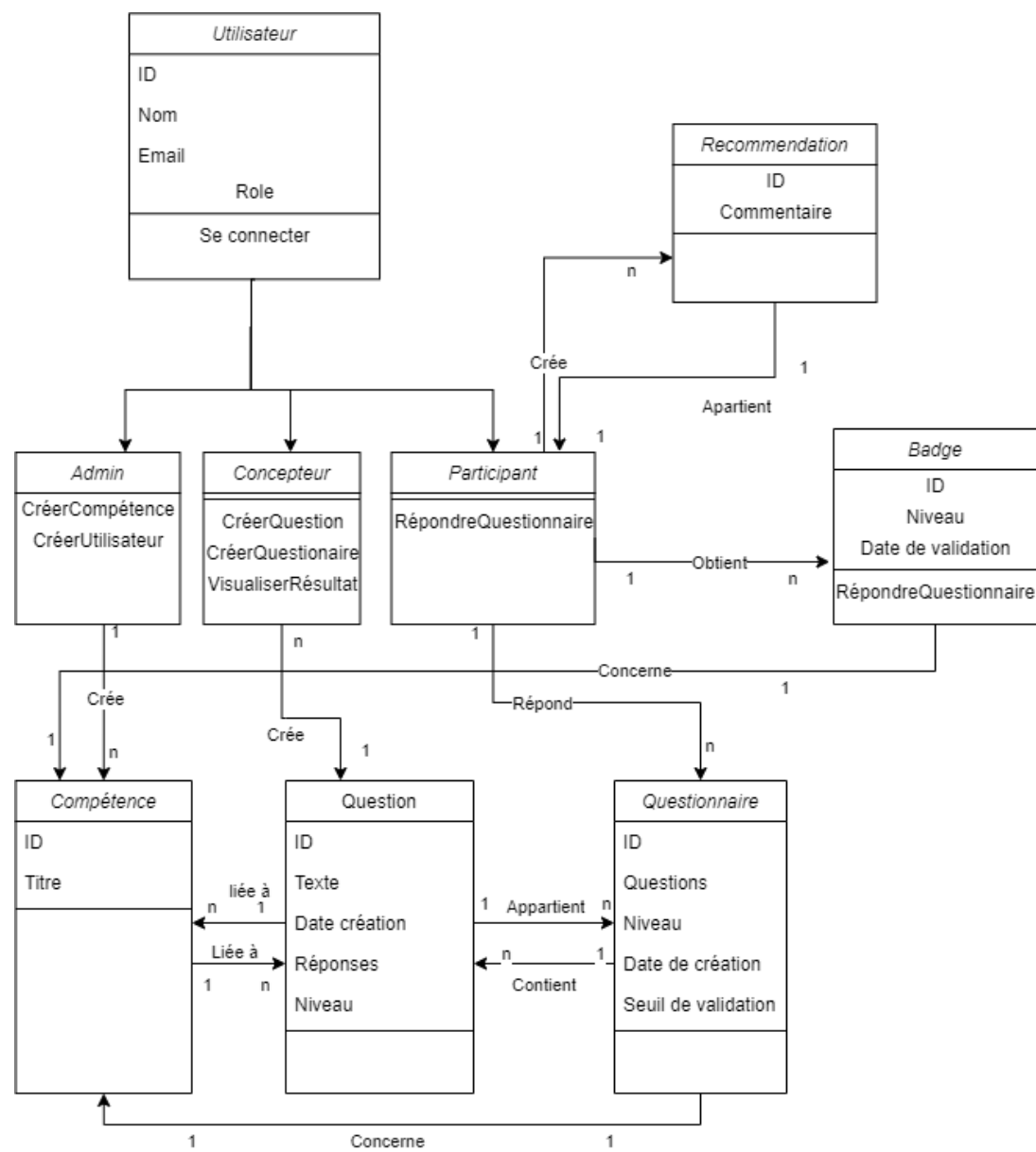


Figure 11 : Diagramme de classe

Notre application se basera sur une architecture « double » MVC (Model-View-Controller) représenté sur la figure ci-dessous. En effet le code est découpé en deux grandes parties. Une première partie est le frontend qui sera codé avec le framework Angular en TypeScript ainsi que du HTML et du CSS. Et la seconde partie est le backend qui sera codé avec le framework Spring en Java.

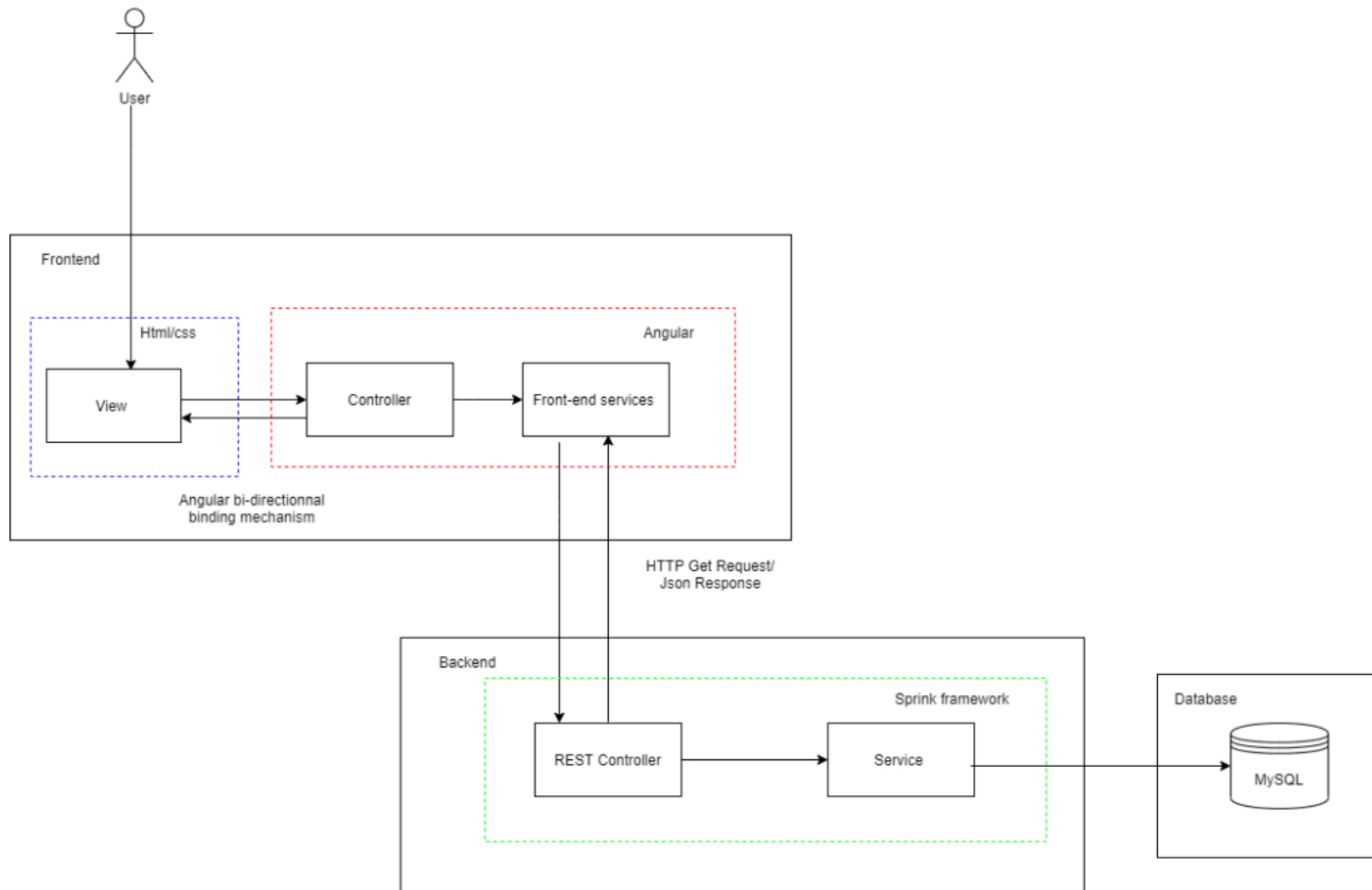


Figure 12 : Diagramme de déploiement