# *RAPPORT DU Brief Step* by Step: Skills Tree

*Youcode Safi*

Projet effectué

À Safi

***A****NNÉE DE LA FORMATION : 2020-2021*

**Développement d’une application JAVAFX avec une Base**

**De données MYSQL**

**Réalisé par :** Choukri Mohamed El Mehdi

Liste des figures

1. Figure 1 : Diagramme de cas d’utilisation 15
2. Figure 2 : Diagramme de séquence 16
3. Figure 3 : Diagramme de classe 17
4. Figure 4 : Technologies utilisées 18
5. Figure 5 : Interface d’Authentification 25
6. Figure 6 : Interface d’inscription 26

**Introduction**

A YouCode, les apprenants sont confrontés à des mises en situation concrètes via des projets en équipe ou individuels. Ce travail collaboratif est très important, car les apprenants sont encouragés à échanger entre eux, quel que soit le type de projet en cours, pour avancer et trouver des réponses. C’est cette synergie entre apprenants qui fait de YouCode une école à part.

La formation à YouCode fait appel à un parcours bien défini destiné à offrir toutes les chances aux étudiants intégrant l’école.

Ouvrir le numérique à tout type de public, tisser un lien avec les acteurs de l’emploi au Maroc et partager les caractéristiques du positionnement pédagogique de YouCode.

Adapter le parcours et la pédagogie aux profils de nos candidats, les informer sur les objectifs et sur leurs résultats au fur et à mesure de la formation.

Immerger les candidats dans un environnement pro en recréant les conditions du monde du travail, impliquer les entreprises nationales pour faire de YouCode un vivier de compétences, assurer la réussite collective des promos grâce à l’appui d’experts techniques, de référents…, évaluer les compétences acquises durant toute la durée de la formation.

Orienter les étudiants vers des emplois adaptés à leur profil, mettre en œuvre les acquis de leur formation en situation professionnelle, lever les freins des entreprises en matière de qualification, évaluer les formations par les entreprises en organisant des rencontres entre les étudiants et les professionnels et mesurer l’impact de la formation YouCode auprès des entreprises.

**Introduction générale sur le projet**

Notre travail a pour but de réaliser une application JAVA permettant la gestion des compétences d’un apprenant et la visualisation des ses derniers par l’apprenant et un staff responsable de la formation.

Pour mener à termes ce projet nous avons dû effectuer des choix techniques, conceptuels et méthodologiques, identifier les différents besoins du projet, réaliser une conception détaillée du projet et enfin réaliser l’application. D'où le présent rapport qui se résume en deux chapitres catalogués comme suit :

* Le premier chapitre comprend l’analyse et la conception de l’application
* Et le deuxième chapitre est destiné aux réalisations et tests de l’application

**Chapitre 1**

Phases de développement

7 étapes dans la vie d’un logiciel / système d’information :

* Planification (Étude de la faisabilité)
* Spécification des besoins (Requirement analysis)
* Analyse (Spécification formelle)
* Conception (Spécification technique)
* Implémentation (Codage)
* Tests unitaires
* Intégration et tests
* Livraison
* Maintenance

Planification

C’est l’activité qui consiste à déterminer et à ordonnancer les tâches du projet, à estimer leurs charges et à déterminer les profils nécessaires à leur réalisation.

Pour notre cas la planification est venue l’or de la demande de notre formatrice qui est le point initial de la planification et mettre en place son besoin à table et à discussion est notre priorité.

On est obligé de suivre ce qui est demandé.

Spécification des besoins

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques et d'identifier les cas d'utilisation initiaux.

**Besoins fonctionnels :**

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une valeur de plus.

Les acteurs en interaction avec notre système

* Apprenant de l’école
* Staff / Formateur

1. Apprenant(s) doit pouvoir assurer les fonctions suivantes :
2. L’inscription
3. L’authentification
4. L’ajout des compétences
5. La modification des compétences
6. La suppression des compétences
7. La visualisation des compétences de cet apprenant
8. Le Staff / formateur a pour tâches principales :
9. L’inscription
10. L’authentification
11. La visualisation des compétences par apprenant

**Besoins non fonctionnels :**

*La performance*: L’application doit être avant tout performante c'est-à-dire à travers ses fonctionnalités, répondre à toutes les exigences des utilisateurs d'une manière optimale.

*La convivialité :*L’application doit être facile à utiliser. En effet, les interfaces utilisateur doivent être conviviales.

Analyse

1. Introduction

Dans ce chapitre nous abordons la partie conception du projet, dans laquelle, nous détaillons les différents éléments de conception, à savoir les classes utilisées dans la conception et les fichiers responsables de la conception de la base de données.

**Conception détaillé**

La conception est la plus importante étape du cycle du développement logiciel. Elle se base essentiellement sur la bonne spécification et l’analyse des besoins.

Notre démarche débute par la compréhension du problème. Ensuite nous analysons le problème pour donner une solution adéquate. A présent, nous sommes dans la phase de concevoir la solution. Notre conception doit obéir à l’architecture déjà choisie pour les différentes parties du système d’où cette phase préliminaire qui nous permettra de définir les composants globaux de notre système.

1. Travail demandé

Contexte :

Dans le but d’informatiser les outils pédagogiques du suivi de l’état d’avancement de chaque apprenant, on propose l’utilisation des arbres de compétences sous format numérique. L’idée est de développer une application desktop qui permet la gestion de l’arbre de compétences. Cette dernière offre la possibilité à chaque apprenant de s’autoévaluer et de visualiser son état d’avancement.

* 1. **Cahier de charge**

**Quoi ?** Notre formatrice nous a proposé de créer une application pour faciliter la gestion des compétences des apprenants (CRUD), les inscriptions et les authentifications des staff / apprenants d’une façon simple et rapide pour aboutir vers la fin à un travail professionnel et éliminer la redondance dans chaque opération.

**Pourquoi ?** Il n’existe pas encore une application Desktop qui traite ce genre de besoin qui sont comme suites :

* + - 1. L’utilisateur de l’application se connecte tant que staff ou apprenant
      2. L’apprenant pourra valider un ou plusieurs niveaux d’une ou plusieurs compétences
      3. Les membres de staff auront la possibilité de sélectionner l’apprenant et de visualiser son état
      4. D’avancement sans pouvoir le modifier
      5. La durée de formation est de deux ans.
      6. Les apprenants de la première année devront valider un référentiel « à préciser !»
      7. Une fois réussi, Les apprenants de la deuxième année devront valider un autre référentiel « à préciser !»
      8. Chaque référentiel est réparti en un nombre fini de compétences et chaque compétence se compose de trois niveaux classés par ordre croissant de difficulté N1, N2 et N3.
      9. La validation de chaque compétence nécessite la validation des trois niveaux
      10. La validation de chaque référentiel nécessite la validation de toutes les compétences
      11. La formation basée sur le principe de la pédagogie active. Chaque apprenant est censé de valider tous les niveaux de toutes les compétences afin de réussir l’année

**Comment ?** L’idée consiste à créer une application liée à une base de données. Cette application va informatiser les tâches effectuées par l’utilisateur.

1. **Solution**

Après avoir connaître la problématique posée et la solution proposée, il est temps de savoir par quoi on doit commencer.

La conception est une étape très primordiale dans chaque projet informatique ;

Que ça soit avec UML ou Merise.

Dans ce projet nous allons choisir le formalisme :

* UML

1. **Définition des termes**

**MySQL** : est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde4, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL server.

**Le diagramme de classe :** Le diagramme de classes est une modélisation statique du système d’information en termes de classes et de relations entre ces classes. Son intérêt réside dans la modélisation des entités du ce système d’information.

**Les cas d’utilisation :** Un diagramme de cas d’utilisation capture le comportement d’un système, d’un sous-système, d’une classe ou d’un composant tel qu’un utilisateur extérieur le voit.

**Le diagramme de séquence :** Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un [Diagramme des cas d'utilisation](https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation). Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

**Diagramme de cas d’utilisation**

Les acteurs et les cas d’utilisations de notre application sont :

* Le Formateur / staff:
  + S’authentifier
  + S’inscrire
  + Visualiser les compétences
* **Signataire :**
  + S’inscrire
  + S’authentifier
  + Ajouter une compétence
  + Modifier une compétence
  + Supprimer une compétence
  + Visualiser les compétences

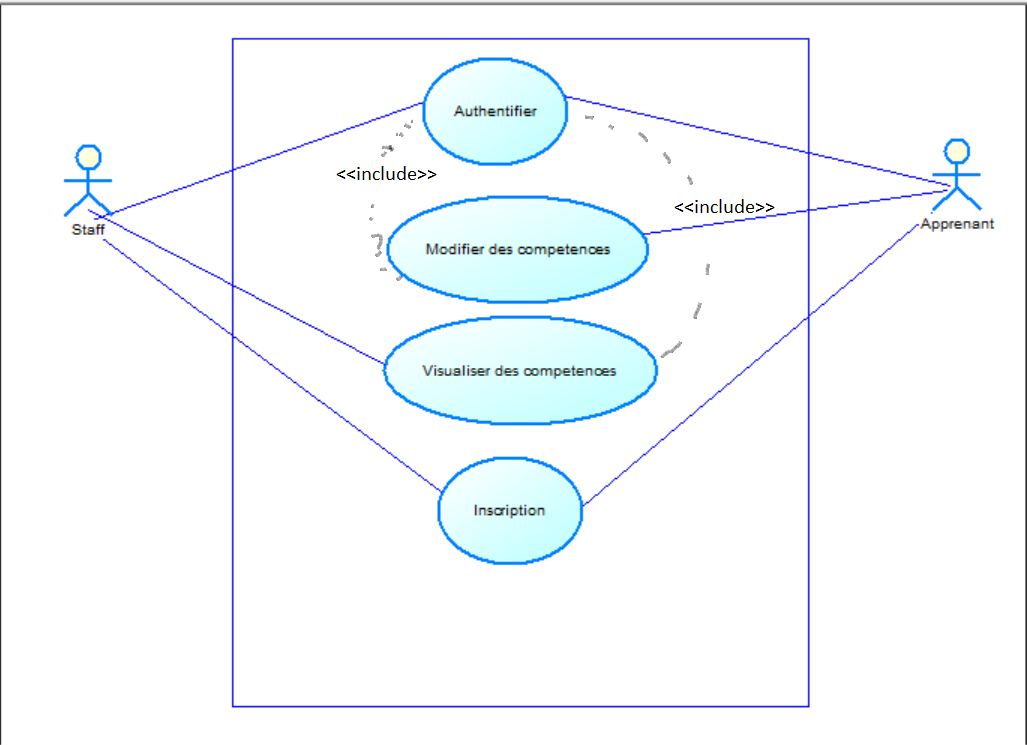


Figure 1 : Diagramme de cas d’utilisation

**Diagramme de séquence**

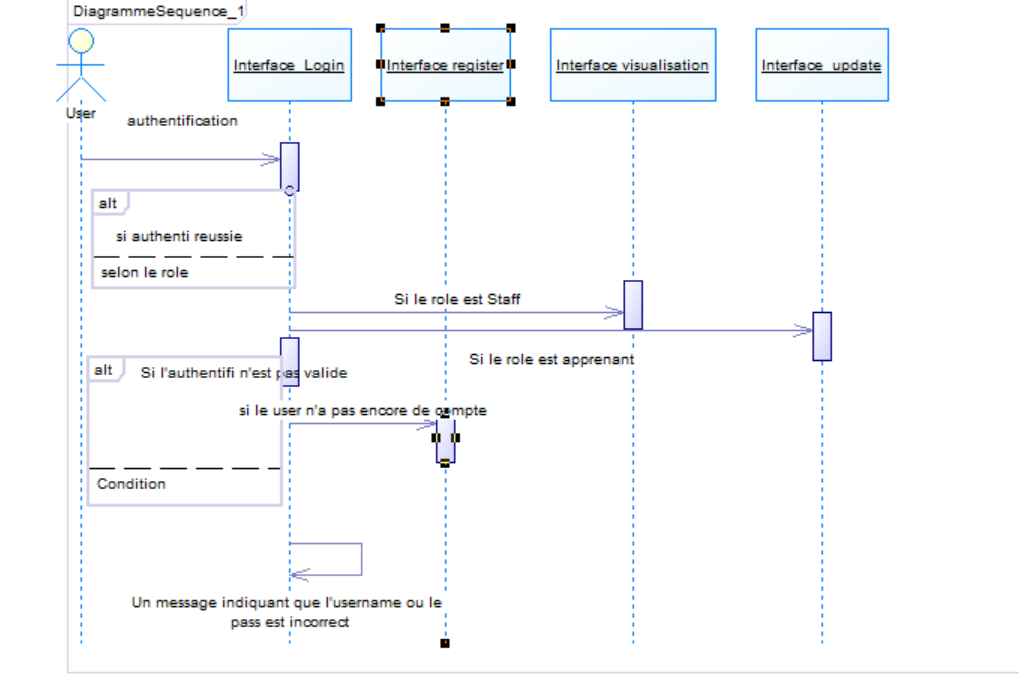


Figure 2 : Diagramme de séquence

**Diagramme de classe**

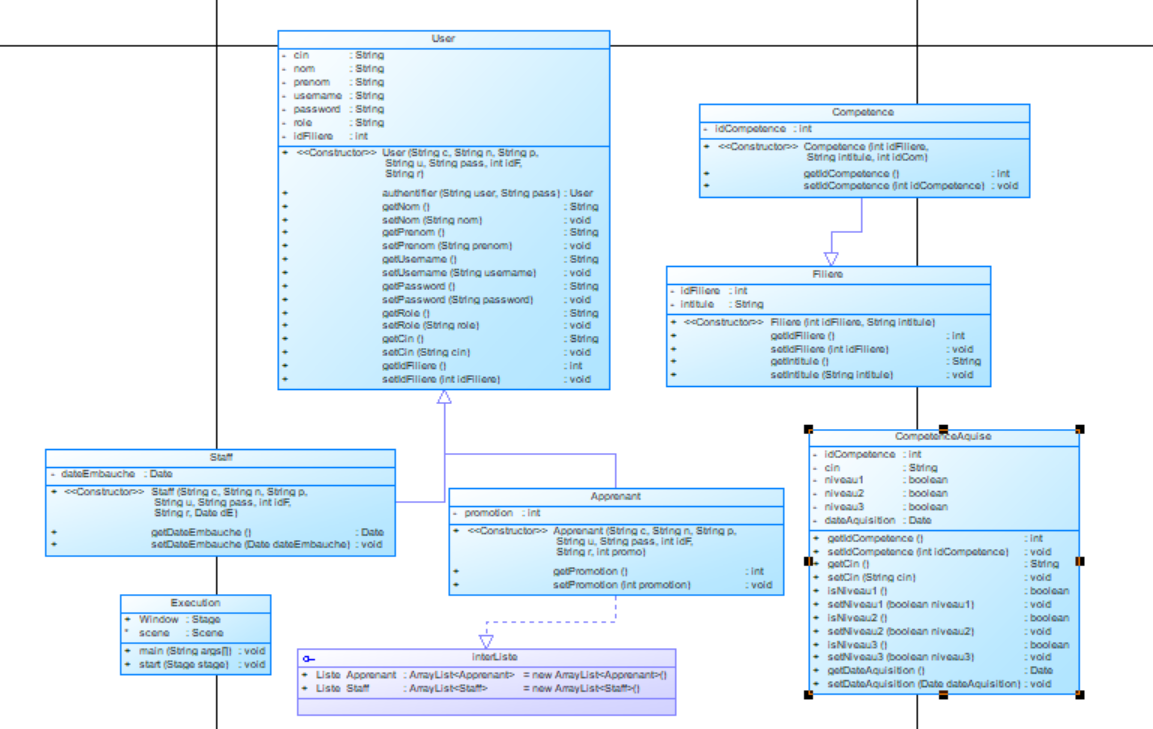


Figure 3 : Diagramme de classe

**Partie II :**

* **Côté programmation :**
* Pour le côté programmation, l’application est programmée en JAVA comme mentionnée au début Elle comporte 7 différentes classes élémentaires (Les tables) et une autre assurant un traitement donné, chacune d’eux assure un rôle diffèrent en collaboration avec les autres.
* (Le digramme de classe précédent explique les relations entre les classes)
* **Côté base de données :**

1. **Introduction**

Pour le côté base de données, l’application utilise MySQL pour stocker les données de tous les compétences, compétences acquises, Utilisateurs et filières ainsi qu’effectuer les différentes transactions sur ces enregistrements en toutes simplicité à savoir l’ajout, la modification et la suppression

**La table :** « user » comporte 9 champs, elle aide l’utilisateur à s’authentifier afin d’accéder à l’application.

|  |  |
| --- | --- |
| **Les champs** | |
| CIN | Clé primaire, elle présente l’identifiant de l’utilisateur. |
| PASSWORD | Elle désigne le mot de passe pour y entrer. |
| ROLE | Elle est le moyen de savoir si la personne authentifier est un Staff ou apprenant |
| USERNAME | Elle désigne le nom complète de l’utilisateur |
| ID\_FILIERE | Est l’identifiant du groupe ou l’utilisateur appartient |
| NOM | Elle désigne le nom de l’utilisateur |
| PRENOM | Elle désigne le prénom de l’utilisateur |
| DATE\_EMBAUCHE | Elle désigne la date d’embauche du staff sinon elle prend la valeur null pour les apprenants. |
| PROMOTION | Elle désigne la date ou l’apprenant a fait l’inscription sinon elle prend la valeur null pour les staffs. |

**La table :** «liste\_staff » comporte 3 champs, elle aide l’administrateur à ajouter les emails des signataires potentiels

|  |  |
| --- | --- |
| **Les champs** | |
| CIN\_STAFF | Clé primaire, identifiant de la table et l’identifiant du staff qui est ajouté |
| ADD\_DATE | Elle désigne la date ou l’opération a été effectuer |
| ADDED\_BY | Elle désigne le CIN de l’administrateur qui a fait l’opération d’ajout de ce staff |

**La table :** «compétence » comporte 3 champs, c’est ici ou on stock toutes les compétences qui existent.

|  |  |
| --- | --- |
| **Les champs** | |
| ID\_COMPETENCE | Clé primaire, identifiant de la table |
| ID\_FILIERE | Elle l’identifiant de la filière ou elle existe cette compétence |
| INTITULE | Elle désigne l’intitulé de la compétence |

**La table :** «competence\_aquise » comporte 5 champs, c’est ici ou on associe chaque compétence a un apprenant.

**NB : ici la clé primaire est la combinaison des deux clés étrangères.**

|  |  |
| --- | --- |
| **Les champs** | |
| ID\_COMPETENCE | Clé étrangère, identifiant de la table |
| CIN | Clé étrangère, elle présente l’identifiant de l’utilisateur. |
| NIVEAU1 | Elle désigne si ce niveau a été compléter ou non 1 = pour vrai et 0 = pour faux |
| NIVEAU12 | Elle désigne si ce niveau a été compléter ou non 1 = pour vrai et 0 = pour faux |
| NIVEAU3 | Elle désigne si ce niveau a été compléter ou non 1 = pour vrai et 0 = pour faux si le niveau 3 est complété la valeur de DATE\_AQUISITION est remplis par la date d’aujourd’hui |
| DATE\_AQUISITION | Elle désigne la date ou le niveau 3 a été finalement compléter |

**La table :** «filière » comporte 2 champs, elle aide l’administrateur à ajouter les groupes potentiels dans la promotion.

|  |  |
| --- | --- |
| **Les champs** | |
| ID\_FILIERE | Clé primaire, identifiant de la table |
| INTITULE | Elle désigne l’intitulé de la filière (groupe dans notre cas) |

**Conclusion**

Dans ce chapitre nous avons détaillé les différentes vues conceptuelles à réaliser à travers le modèle UML, ainsi que la conception de la base de données. Cette conception est essentielle pour la phase de réalisation qui constitue l’objet du chapitre suivant.

Chapitre 2

**REALISATION ET TESTS**

**Réalisation et tests**

1. **Introduction**

Dans ce chapitre, nous nous intéressons à la description de quelques interfaces et difficulté rencontré l’or de l’implémentation du système.

1. **Interfaces de l’application**

Nous exposerons quelques interfaces de notre application.

Lors du lancement de l’application une interface d’authentification s’affiche.

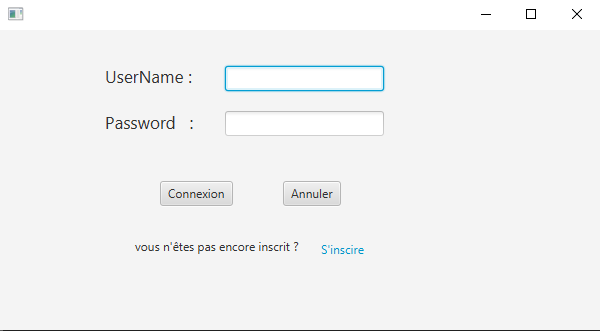


Figure 4 : Interface d’Authentification

Cette interface a pour but de faire l’authentification des différents l’utilisateur

* Chaque utilisateur doit s’identifier par son USERNAME pour accéder au contenu de l’application qui est diffèrent selon le type de ce dernier
* Si l’utilisateur n’est pas encore inscrit il est donc amené à cliquer le lien en bleu pour aller s’inscrire

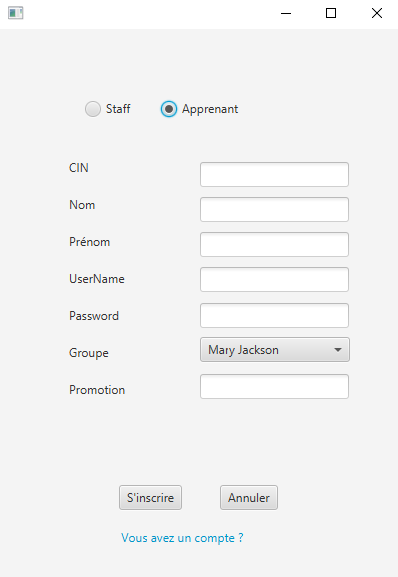


Figure 5 : Interface d’inscription

* Concernant l’inscription l’utilisateur (staff) doit être ajouter par l’admin pour qu’il puisse effectuer l’inscription sinon ça demande est rejetée sinon l’inscription est tout à fait normale pour l’apprenant.
* L’inscription est également interdite pour un utilisateur existant dans la base de données.

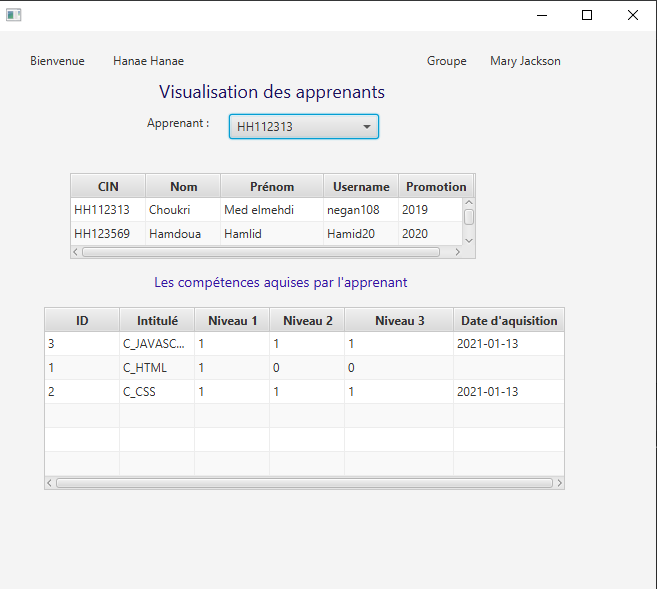


Figure 6 : Interface qui suit l’authentification pour Staff

* Le comboBox apprenant est remplis par les CIN des apprenants qui sont dans la même classe que le formateur / staff.
* Au-dessus du combobox il existe la TableView qui contient tous les apprenants qui sont dans la même classe que le formateur.
* Une fois un élément est sélectionné du combobox la TableView Compétence est remplis par les compétences acquises de l’apprenant sélectionné
* Une formule de bienvenue suivie du nom de l’utilisateur.
* Ici le formateur / staff ne peut que visualiser l’état d’avancement des apprenants.

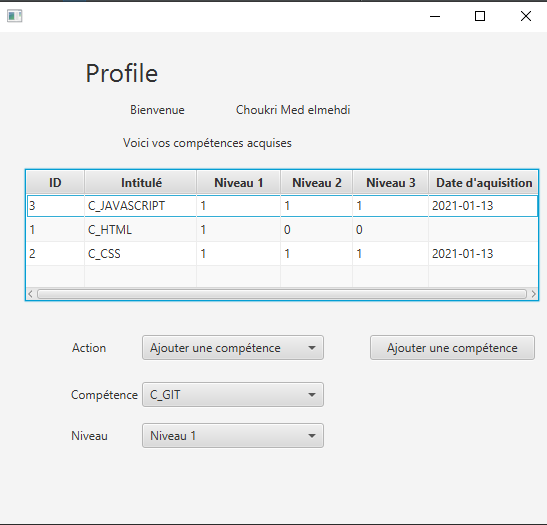


Figure 7 : Interface qui suit l’authentification pour Apprenant

* Le comboBox Compétence est remplis par l’intitulés des compétences qui existent dans la table compétence et selon chaque action le comboBox charge différemment .
* En-dessus de la TableView il existe le comboBox qui contient les actions qu’un apprenant peut faire à savoir l’ajout d’un compétence la modification de la compétence et enfin la suppression de la compétence.
* La TableView Compétence est remplis par les compétences acquises par cet apprenant
* Une formule de bienvenue suivie du nom de l’utilisateur.
* Ici l’apprenant visualiser son état d’avancement modifie ainsi ses compétences.