Réalisation d'une application JakartaEE pour la gestion des formations (v1)

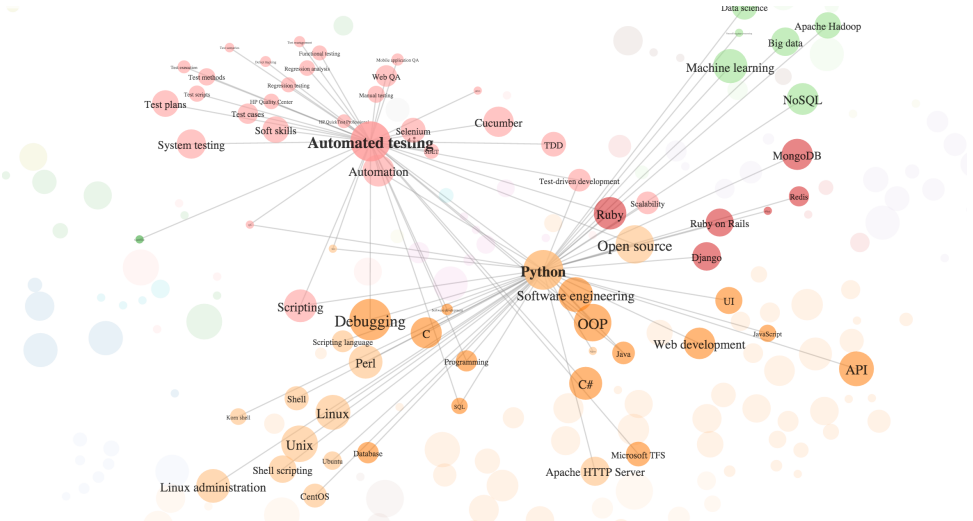
Dans le cadre du cours Développement d'Applications Web 2021/2022
Pour tout éclaircissement, contacter Massinissa Hamidi (hamidi@lipn.univ-paris13.fr) avec pour objet [insset-l3mn]

L'idée de ce projet est de permettre aux apprentis d'un domaine de votre choix (préférablement proche de vos centres d'intérêt) de pouvoir être évalués sur leurs compétences et d'accéder à des parcours de formation personnalisés générés automatiquement. Dans ce qui suit nous allons décrire les principaux composants de l'application à développer illustrées sur le domaine de l'informatique. Néanmois, il est obligatoire que vous choisissiez des domaines distincts. Ces spécifications sont susceptibles d'être rafinées. D'ailleurs nous vous encourageons à le faire.

Le projet est à réaliser sous forme d'une application JakartaEE. Il y a un effort de modélisation à faire dans un premier temps, notamment pour le schéma de la base de données avec la notion de graphe de compétences (détaillée dans la suite). Dans un second temps, il faut concevoir des Jakarta RESTful Web Services.
Les développements seront à réaliser sur des dépôts GitHub qui vous seront crées prochainement. Des rendus intermédiaires vous seront demandés jusqu'au rendu final.
D'autres éléments, tels qu'un petit compte rendu et éventuellement une vidéo courte (~3 minutes) pouront vous être demandés à des fins de documentation.

Graphe des compétences

Les compétences dans le domaine de l'informatique sont organisées sous forme de graphe orienté, i.e., certaines compétences sont des pré-requis et la validation d'une compétence est conditionnée par la validation ou la satisfaction de ces pré-requis. Ci-après un exemple de graphe des interconnexions entre un certain nombre de compétences techniques[1].



[1] https://insights.dice.com/2015/07/01/dice-data-how-tech-skills-connect/

On aura trois types de graphe :

- un graphe contenant des données communes à tout le monde ;
- un graphe propre à chaque utilisateur ;
- un graphe produit à partir des réponses aux questions posées permettant de mesurer les compétences d'une personne.

Gestion des apprenants

Un apprenant doit créer un compte dans le système avant de pouvoir l'utiliser. Après inscription, il accède au système en s'identifiant.
Il pourra ensuite entamer une procédure qui permet de déterminer son niveau en informatique : d'abord, une série de questions élémentaires devront être posées à l'apprenti ; les réponses à ces questions déterminent les paramètres (niveau, difficulté, domaine visé--e.g. data engineer, web developer--, etc.) qui permettront de générer le questionnaire d'évaluation ; enfin, le questionnaire d'évaluation lui est soumis et un niveau lui est assigné.
La notion de niveau, ici, pourrait correspondre au graphe de compétence, ci-haut, dans lequel les nœuds seraient pondérés par le score que l'apprenti aurait obtenu dans la compétence correspondante.
Selon le niveau obtenu, le système lui propose un parcours de formation personnalisé afin d'atteindre le niveau requis dans le domaine souhaité.

Génération de questionnaires

Afin de déterminer le niveau d'un apprenti, le système génère un questionnaire qui doit cerner son niveau en déterminant ses compétences. Si le système n'est pas certain du niveau de l'apprenant dans une compétence donnée, le système régénère des questionnaires supplémentaires (plus affinés) jusqu'à ce que le système soit certain des compétences de l'apprenant. Évidemment, l'objectif est de déterminer le niveau de l'apprenti rapidement.

Génération des parcours de formation

En fonction du niveau obtenu par l'apprenti, le système propose des formations personnalisées. Ces formations peuvent être construites à partir du graphe de compétences.

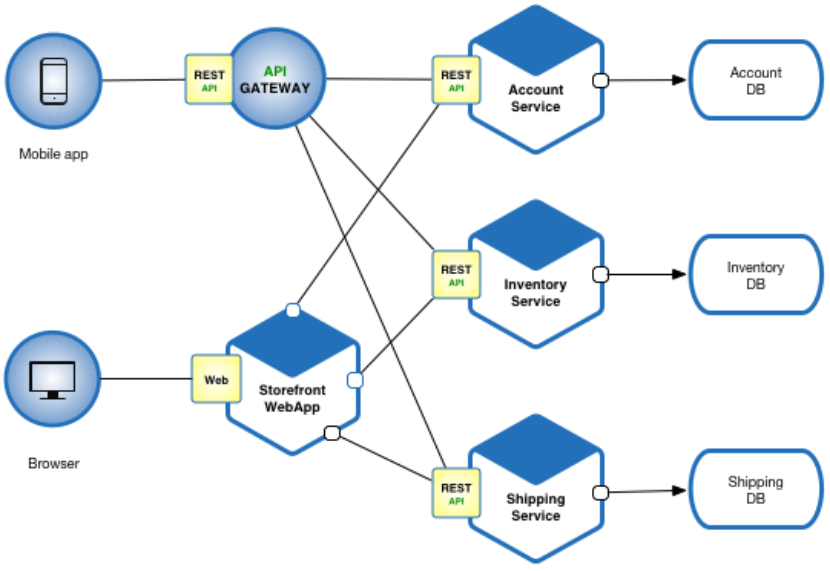
Gestion des formateurs

Les formateurs peuvent ajouter des compétences, rédiger des questions, modifier des questions, etc. Le système pourra aussi leur assigner les parcours des apprentis dans une interface graphique bien organisée afin de vérifier la cohérence des questionnaire et du contenu des parcours eux-même.
Les formateurs peuvent aussi soumettre des suggestions de modification du graphe de compétences : ils peuvent rajouter des compétences et les lier à celles déjà présentes dans

le graphe ; ils peuvent en supprimer certaines, etc. Les modifications doivent être commentées (ou argumentées).
Celles-ci restent des suggestions de modification, car le graphe de compétence principal qui est utilisé dans le système est unique et est maintenu par un ou plusieurs administrateurs à qui incombe la responsabilité de le mettre à jour. Il faut à cet effet, mettre en place un système qui maintient une trace des différentes versions du graphe et des raisons de sa modification.
L'interface devra permettre de construire et modifier le graphe de compétence ainsi que naviguer dans son historique de modifications.

Mise à jour 16/02/2022

- Avant toute chose, gardez en tête que la première version de votre modélisation vous permettra d'y voir plus claire. Le but n'est pas d'avoir quelque chose de parfait dès le départ mais simplement un premier jet sur lequel on pourra raisonner ensemble lors de nos prochaines séances.
- Je tiens aussi à préciser que chaque groupe choisit un unique domaine de formation obligatoirement différent de ceux choisis par les autre groupes. Je rappelle qu'un domaine de formation c'est par exemple "Informatique", "Mécanique", "Football", "Hôtellerie", etc.
- Notez aussi que pour vous familiariser avec des aspects architecturaux, il faut faire l'effort de subdiviser la partie back-end en microservices. Autrement dit, au lieu d'avoir une seule et unique application web JakartaEE en backend (.war) dite monolitique, il faut en avoir une pour chaque composant indépendant de votre application (cf. image ci-dessous). Par exemple, une pour la gestion des utilisateurs, une autre pour la gestion des graphes de connaissances, une pour la génération des évaluations (ou questionnaires), etc. Cela peut se faire dans un deuxième temps et on aura l'occasion d'en rediscuter ensemble. Voici entre-temps un article interessant à ce sujet
https://microservices.io/patterns/microservices.html



Réponses aux questions

Dans la suite, quelques réponses aux questions posées sur Discord. Vous remarquerez l'utilisation du terme “imagniation” : c'est ce que le format du projet vous pousse à faire.

- L'identifiant correspond à une adresse de courriel ou à un pseudo ?
- Un apprenti s'inscrit lui-même ou via une invitation d'un formateur ?

Ici, choisissez la méthode que vous estimez la plus appropriée.

- Un formateur A peut-il éditer les questions d'un formateur B ?
- Un formateur A peut-il éditer les données d'un apprenti d'un formateur B ?

Si vous considérez ce cas, vous pouvez imagnier une catégorisation des formateurs selon les compétences dont ils ont la charge. Par exemple, les formateurs en “Pâtisserie fine”, “Intelligence artificielle”, “Symfony”, etc. Les formateurs d'une même compétence peuvent mutuellement se suggérer des modifications. Seuls les formateurs en “Symfony” peuvent proposer une suggestion de modification aux formateurs en “Symfony”. Le tout est de maintenir un historique des suggestions et des modifications.

- Le domaine choisi lors de la "série de questions élémentaires" ne défini-t-il pas les questions à poser ? Le graphe est-il donc différent entre chaque domaine sélectionné lors des questions élémentaires ? Est-ce des sous-graphes intégrés au sein d'un gros graphe (des parties seront donc inutiles pour un apprenti A, B, C ...) ?
- Avez-vous la liste exacte des actions possibles d'un formateur ?

La série de questions élémentaires doit se focaliser sur la ou les compétences visées. L'idée est que l'apprenant choisisse une ou plusieurs compétence(s) qu'il souhaite acquérir dans le domaine de formation et la série de questions vise à déterminer son niveau dans les compétences requises par celle(s)-ci. À vous d'imaginer jusqu'à quel point dans les pré-requis vous devez aller afin de piocher et générer des questions.

- Cela ne revient-il pas au même ? Puisque chaque utilisateur à un graphe forcément lié aux réponses ?

Le graphe lié aux réponses correspondrait aux réponses données lors d'une session d'évaluation donnée (à un temps t et dans un contexte particulier). Ça serait un sous-graphe puisque focalisé sur un sous-ensemble des compétences.
Le graphe de l'utilisateur correspondrait à son évolution au fur et à mesure des formations qu'il suit.

- React et Jakarta dans un même dépôt ?

L'idée est qu'au final vous ayez une application fonctionnelle avec les deux parties font-end (Framework Javascript : react, angular, vue, éventuellement réalité augmentée) et le back-end (JakartaEE).

- Chaque question est lié à une/plusieurs compétences ? Ou une question s'applique seulement à une seule compétence ?

- Avez-vous une liste de ces questions et de leur valeur en point ? Afin de tester les schémas et de faire des tests ensuite ?
- Si l'app n'est pas certaine du niveau de l'apprenant et qu'aucune question supplémentaire n'est disponible, quelle action doit prendre l'app ?
Il faut différencier entre la modélisation des questionnaires d'évaluation et leur génération. On peut imaginer, même si cela ne vous est pas demandé, un composant basé sur une intelligence artificielle capable de générer des questionnaires à l'infinie. Vous pouvez éventeullement faire cette supposition.

- On est sur quel type de questionnaire ? Oui/Non ? QCM ? Les deux ?
Les deux mais aussi des réponses libres (ce qui peut ouvrir la voie à du traitement automatique de la langue même si on vous demande pas d'aller aussi loin).
J'ai restreint le type d'évaluation dans l'énoncé aux QCM mais comme j'ai pu l'indiquer à certains d'entre vous, vous pouvez considérer une forme beaucoup plus abstraite. Vous pouvez par exemple imaginer ce qui peut se faire dans un domaine de formation lié au sport où la notion d'évaluation ne se limite pas à des questionnaires mais inclut également des tests physiques ou techniques (niveau balle au pieds ou précision des coup-francs, dans le cas du football par exemple).

- Un apprenant peut-il perdre des points sur une questions si sa réponse est incorrecte ?
Pas nécessairement, mais vous pouvez imaginer un système qui détecte les incohérences. Exemple : il répond juste à une question mais se trompe sur une autre question liée (ou même question reformulée différemment)

- Vous avez écrit
"L'idée de ce projet est de permettre aux apprentis d'un domaine de votre choix"
puis
"Néanmois, il est obligatoire que vous choisissiez des domaines distincts."
puis
"les réponses à ces questions déterminent les paramètres (niveau, difficulté, domaine visé--e.g. data engineer, web developer--, etc.)"
---> On doit choisir un domaine, différent entre chaque groupe ? Mais en même temps l'appli doit en proposer plusieurs ???
Chaque groupe choisit un unique domaine de formation obligatoirement différent de ceux choisis par les autre groupes. Je rappelle qu'un domaine de formation c'est par exemple "Informatique", "Mécanique", "Football", "Hotellerie", etc.

- Vous avez écrit :
"un graphe contenant des données communes à tout le monde ;"
--> Ce graphe correspond à un graphe des réponses de tout le monde ? ET/OU un graphe lié à la config de l'admin et donc vide de réponse/valuation ?
Initialement, on entendait plutôt un graphe global du domaine comme celui présenté dans la figure dans le cas du domaine informatique. Il est construit par des experts de ce domaine.
Vous pouvez tout de même imaginer un graphe qui syntétisant la connaissance de tous les apprenants du système ou d'une promotion afin de visualiser le niveau global.

- On descend jusqu'à quel niveau dans l'iceberg ? (adminsys -> Apache et stop ? pour adminsys -> Apache -> config -> reverse proxy -> ... ? Pas de limite ?)

Descendre au niveau qui vous parrait nécessaire pour les besoins de démonstration de votre application. Vous pouvez eventuellement détailler le plus possible une partie du graphe à des fins de démonstration ou de teste.

- Vous avez écrit :
"Si le système n'est pas certain du niveau de l'apprenant dans une compétence donnée"
--> Cela veut dire quoi ? L'apprenti à soit bon, soit mauvais ? Il y a désormais une superposition d'état lol ? Si dans le domaine "webdev", compétence "BDD", je réponds mal à une question sur SQL, j'aurais forcément une proposition de révision en SQL ? Ou pas suivant le niveau global de la compétence ? Ou sur la compétence globalement ? Comment définir cela ?
Effectivement, dans le cas où les réponses aux questions sont déterminisites comme dans pour les QCM. Par contre, si vous considérez une définition plus abstraite des évaluations/questionnaires, comme discuté plus haut, vous n'avez plus cette certitude. On peut avoir par exemple un bout de code dont l'évaluation n'est pas forcément liée seulement au fait que le resultat soit juste. Vous pouvez imaginer dans le cas précis de SQL une évaluation selon la compléxité de la requête fournie par l'apprenant avec par exemple le nombre de jointures effectuées, le temps nécessaire pour retourner une réponse, etc.

- Vous avez écrit :
"les nœuds seraient pondérés par le score que l'apprenti aurait obtenu dans la compétence correspondante."
--> Compétence PHP, sous-compétence Symfony et sous-compétence Lavarel : les questions pour une compétence ont-elles le même "coeff" ? Par exemple Symfony peut-il plus influer le niveau que Lavarel pour la compétence PHP ? Paramétrage par le formateur ou l'admin ?
Cette information devrait être encodée dans le graphe global du domaine. Pour des besoins de démonstration, essayez d'avoir quelque chose qui ressemble plus ou moins à la réalité même si cela reste subjectif.