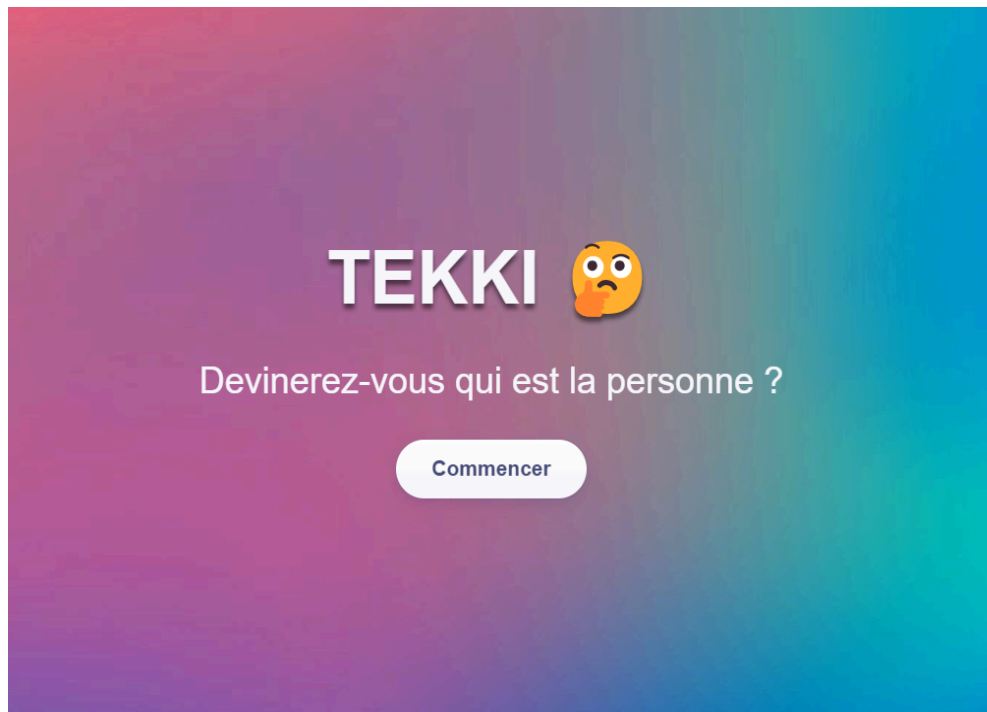


Rapport du projet Génie Logiciel

Sujet du projet : Jeu de devinette - TEKKI

21/05/2024



Réalisé par :

Max PENSO

Younes OUAMAR

Elouan HAMON

Lakshan DISSANAYAKE

Hai-My LE

SOMMAIRE

1. Vue d'ensemble

- 1.1. Descriptif du projet à réaliser
- 1.2. Technologies utilisées

2. Contexte de travail

- 2.1. Travail réalisé
- 2.2. Difficultés rencontrées

3. Bilan

1. Vue d'ensemble

1.1. Descriptif du projet à réaliser

Le jeu **Tekki** est inspiré du célèbre jeu "Qui est-ce?". L'objectif du jeu est d'identifier une personne en se basant sur des indices fournis par l'ordinateur. Les joueurs commencent avec un certain nombre de points et peuvent poser des questions pour affiner leurs suppositions ou faire une devinette directe.

Au début du jeu, le joueur commence avec 150 points. L'ordinateur propose une liste de questions que le joueur peut poser pour obtenir des indices sur la personne à deviner. Chaque question posée coûte 10 points au joueur. En réponse à chaque question, l'ordinateur fournit une information utile pour aider le joueur à identifier la personne. À tout moment, le joueur peut choisir de deviner directement sans poser de questions supplémentaires. Si le joueur décide de deviner, une liste de personnes possibles lui est proposée, et il doit sélectionner celle qu'il pense être la bonne. Si le joueur devine correctement, le jeu se termine et les points restants représentent son score final. En cas de mauvaise devinette, 20 points sont retirés de son total, mais il peut continuer à poser des questions ou faire une nouvelle tentative de devinette. Ce système permet au joueur de stratégiquement gérer ses points tout en cherchant à identifier la personne correcte le plus rapidement possible.

1.2. Technologies utilisés

Nous avons organisé notre projet en utilisant le modèle MVC (Modèle-Vue-Contrôleur) ainsi que des repositories pour bien séparer les différentes structures. Le modèle a été développé en Java, le contrôleur a été implémenté avec Spring Framework, et la vue a été gérée avec Thymeleaf. Pour les tests, nous avons utilisé Mockito et JUnit 5 pour garantir la fiabilité et la robustesse de notre application.

Nous avons fait le choix d'utiliser les sessions spring grâce à l'annotation `@SessionAttributes` ce qui nous permet de handle facilement les parties, le score, les personnes sélectionnées et les questions. Nous avons aussi fait le choix de gérer le score dans `MainController`.

2. Contexte de travail

2.1. Travail réalisé

Dans le dossier Java de notre projet, nous avons les composants suivants :

AppConfig : Configuration de l'application Spring.

MainController : Contrôleur principal gérant les interactions entre la vue et le modèle.

CsvPeopleRepository : Repository pour la gestion des données des personnes à partir d'un fichier CSV.

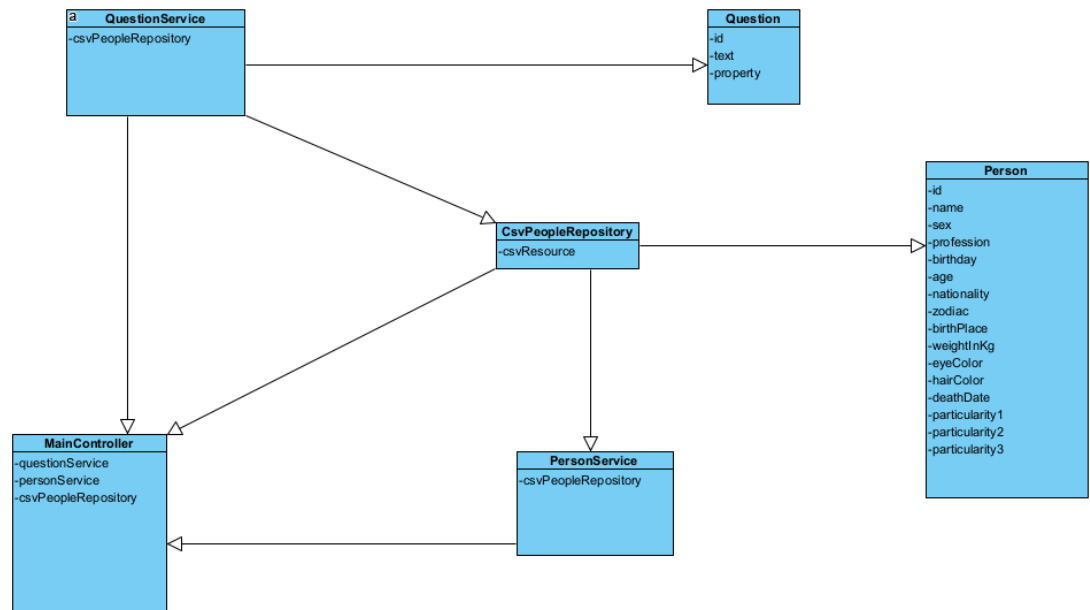
Person : Classe représentant les personnes à deviner.

PersonService : Service gérant la logique métier liée aux personnes.

Question : Classe représentant les questions que le joueur peut poser.

QuestionService : Service gérant la logique métier liée aux questions.

TekkiApplication : Classe principale pour lancer l'application Spring Boot.



À la création des classes, nous avons pu apprendre et découvrir les records en Java, une nouvelle fonctionnalité qui simplifie la création de classes immuables et réduit le code boilerplate. Cela a enrichi notre compréhension de Java et nous a permis d'écrire du code plus concis et lisible.

Durant la réalisation des fonctionnalités un autre travail était opéré, lui aussi très important, les tests. Tout au long de la durée du projet nous testions au fur et à mesure les fonctionnalités réalisées par une autre partie du groupe. Cela nous a permis d'identifier plusieurs problèmes sur des méthodes spécifiques et donc de nous faire gagner du temps car grâce à Maven il été assez facile de comprendre d'où venait les problèmes. Les tests nous permettent aussi de conserver une certaine

crédibilité autour du projet, en effet grâce à eux nous pouvons garantir le bon fonctionnement de chaque méthode.

Concernant le **jeu de données**, nous avons dû réfléchir à son nombre de variables et à la pertinence de ces dernières, nous avons donc retenu 16 colonnes utilisables dans notre jeu qui serviront à filtrer les propositions possibles. Nous avons également discuté du nombre de lignes du jeu de données, initialement aux alentours de 1700 lignes et centré sur des célébrités françaises vivantes et défunes, nous trouvions trop difficile de deviner une bonne partie des personnes présentes dans notre base. Il a donc été décidé de revoir drastiquement la population de célébrités à exploiter, nous sommes donc passés à une centaine de personnalités internationales plus simples à trouver.

Nous avons donc web scrap notre propre jeu de données en nous basant sur le site "<https://hwceleb.com>" via un script python utilisant les bibliothèques Pandas, Requests et BeautifulSoup4. Nous avons ensuite retraité les résultats obtenus en créant de nouvelles colonnes, comme par exemple pour le signe astrologique calculé à partir de la date de naissance obtenue.

Concernant les particularités des personnalités, qui font office d'indice, il n'a malheureusement pas été possible de les obtenir en utilisant un script. Il nous a donc fallu rechercher, sélectionner et renseigner manuellement ces informations.

2.2. Difficultés rencontrées

L'une des principales difficultés de développement a été la gestion de session et l'interactivité entre le front et le back.

En effet, habitués aux frameworks Javascript modernes comme React, Vue et Svelte, nous avons dû revenir à un modèle MVC plus classique, et bien que Spring nous ait facilité la tâche, l'adaptation a été compliquée au démarrage.

Nous avons dû apprendre à utiliser les fonctionnalités natives de Spring et nous renseigner sur la documentation de celui-ci pour ne pas réinventer la roue, notamment sur la gestion des éléments de session pour les ajouter, les appeler et les supprimer au bon moment sans perte d'information en changeant de page.

La principale difficulté pour la rédaction des tests fut l'adaptation aux nombreux changements dans le code, il est arrivé plusieurs fois que la réflexion autour de la rédaction d'une méthode ne soit pas la bonne ce qui a amené à devoir repenser les tests pour qu'il s'adapte à ce nouveau fonctionnement. Une autre difficulté a été l'utilisation de Mockito, son utilisation a été compliquée à comprendre et il est d'ailleurs possible que cette technologie ai encore mal été comprise par les membres de l'équipe ayant effectué les tests, c'est cette difficulté qui nous a empêché de tester efficacement la méthode

Enfin, la dernière difficulté et pas des moindres a été la gestion de git. Au début du projet, nous n'arrivions pas du tout à nous accorder, les push et les pull étaient aléatoires et nous n'avions pas de conventions de nommage (anglais ou français, espaces ou pas). Au fur et à mesure du projet, nous avons réussi à nous adapter les uns aux autres et à avoir une logique plus cohérente.

3. Bilan

3.1. Sommes nous satisfaits ?

Oui, le site, bien que modeste, est pleinement fonctionnel et le cahier des charges a été respecté.

Nous nous sommes surpris à apprécier jouer des parties car les parties sont parfois compliquées et drôles à la fois. Nous l'avons fait essayer à notre entourage qui eux aussi ont apprécié le projet.

3.2. Retour du groupe sur le module :

En général nous avons apprécié ce cours même si dans la globalité du groupe nous n'apprécions pas le génie logiciel. Le rythme du cours était relativement plaisant et nous a permis de nous intéresser à la majorité des sujets.

Si c'était possible de coder et d'améliorer les codes ensemble, cela rendrait l'apprentissage encore plus attractif et pour les étudiants.

3.3. Utilisation des logiciels d'IA :

Nous avons utilisé ces outils, cela nous a permis d'accélérer certains processus comme des parties de code assez répétitives. Cela nous a permis de gagner du temps et lorsque nous rencontrions des problèmes de codage ou des bogues que

nous avions du mal à résoudre, les logiciels d'IA se sont révélés être un outil très utile. Grâce à ses suggestions et explications, nous avons pu comprendre et corriger les erreurs plus rapidement.