Rapport
Projet Services Web
Issam Toure & Edouardo Rodrigues

Introduction

La société EiffelTutoringSolutions s'est engagée dans la création d'une plateforme de tutorat au sein de l'Université Gustave Eiffel, répondant à une demande croissante d'assistance académique parmi les étudiants. Dans le cadre de ce projet, notre mission à consisté à concevoir et implémenter une application Java distribuée reposant sur le middleware Java RMI.

Dans un souci d'expansion, EiffelTutoringSolutions à souhaité déployer la plateforme au-delà des frontières de l'université en intégrant un service Web, dénommé GustaveTutorService. Notre mission, à été donc de déployer ce service, permettant aux personnes extérieures de tirer parti des fonctionnalités offertes. L'application inclus des fonctionnalités de paiement pour ces utilisateurs, avec la possibilité de régler les frais dans la devise préférée de celui-ci, en utilisant les taux de change en temps réel. Pour assurer la transaction financière, le GustaveTutorService interagie avec un service Web Banque afin de vérifier la disponibilité des fonds nécessaires et d'effectuer le paiement dans la devise choisie.

Ce rapport détaillera les choix de conception pris lors du développement de cette application, exposera les défis rencontrés au cours de ce processus, et fournira un manuel d'utilisation avec notamment les fonctionnalités disponibles, pour garantir une expérience utilisateur optimale.

Choix de conception

La réalisation d'une application aussi complexe requiert des choix de conception réfléchis afin de garantir une expérience utilisateur optimale. Notre architecture se déploie autour de trois axes majeurs : le serveur RMI, EiffelTutoringSolutions, le service web, GustaveTutorService, et le service bancaire, Banque.

Pour ce qui du processus d'implémentation de cette application, nous avons opté pour le langage de programmation Java. Le serveur RMI tient une position centrale au sein de ces composants, agissant comme un orchestrateur entre les tuteurs et les étudiants au sein de l'université Gustave Eiffel. Les informations relatives aux tuteurs inscrits et aux rendez-vous, sont stockées dans le serveur RMI en utilisant le type List de Java. Ces fonctionnalités sont exposées par le serveur RMI via l'utilisation de l'option "rebind", tandis que le client y accède à l'aide de l'opération "lookup".

Dans une phase ultérieure, l'objectif était d'étendre ces fonctionnalités du service RMI au-delà des frontières de l'université, afin d'inclure des utilisateurs non affiliés à Gustave Eiffel. Le service, par le biais d'un "lookup", récupère les fonctionnalités du serveur RMI, permettant ainsi de fournir les mêmes services tout en intégrant une fonctionnalité de paiement. Cette fonctionnalité de paiement s'appuie sur le service bancaire, qui fournit toutes les informations nécessaires sur les comptes bancaires à GustaveTutorService pour effectuer des paiements pour le client concerné. Le service bancaire est donc présent coté client et serveur, grâce à la définition du scope « application ».

Une personne extérieure peut donc solliciter ce service pour un rendez-vous avec un tuteur, obtenant ce rendez-vous s'il dispose des fonds nécessaires dans son compte et si le tuteur est disponible. Dans le cas contraire, il est invité à reprogrammer un rendez-vous, ultérieurement. En effet, il est important de noter qu'une personne extérieure, n'étant pas étudiant à l'université Gustave Eiffel, ne bénéficie pas du privilège d'être placée en liste d'attente, au contraire d’un étudiant de l'université Gustave Eiffel. En ce qui concerne les paiements en plusieurs devises, cela dépend du compte en banque du client, assurant ainsi que le paiement est effectué dans la devise associée au compte du client, sans impact sur le choix de devise souhaité, avec un taux de change en temps réel, à l’aide du service FxTop.

Difficultés rencontrés

Lors de la conception de cette application, la principale difficulté à laquelle nous avons été confrontés était d'établir la communication entre le serveur RMI et le service web. En effet, les méthodes présentes dans le serveur web, devaient être conformes aux normes de déploiement d'un service web. Cette exigence a entraîné la nécessité de convertir l'ensemble des méthodes du serveur RMI, une tâche qui a demandé un investissement important en termes de temps et de travail. L'objectif était de garantir une communication fluide entre les deux parties sans aucune perte de données.

Manuel utilisateur

Inscription d'un tuteur

Un tuteur, pour pouvoir proposer ses services, doit s'inscrire sur l'application, et définir ses différentes infos, à savoir, son nom, ses domaines, ses horaires, ainsi que son tarif horaire. A partir de la, son calendrier se remplira des rendez-vous à effectuer.

Recherche de tuteur

Il est possible, pour tout client de rechercher des tuteurs, que ce soit en filtrant ou non. Le client à la possibilité de filtrer par nom, domaine d'expertise, horaires, ainsi que le tarif.

Prise de rendez-vous

Il est possible de prendre rendez-vous avec un tuteur, après avoir réussi à chercher celui-ci. Pour un étudiant de l'université Gustave Eiffel, si le tuteur n'est pas disponible, il sera placé en liste d'attente, et alerté quand son rendez-vous est programmer.

Paiement

Les rendez-vous sont payant pour les personnes venant de l'extérieur, c'est à dire par le service web. Le client à la possibilité dans la devise de son choix, c'est à dire par rapport à son compte bancaire détenue dans le service Banque.

Scénario

Voici le scénario prévu pour la soutenance, à noter que pour les rendez-vous les numéro de jour et années n’ont pas été retenues, pour faciliter la démonstration, nous avons seulement utilisés les jours (les numéros de jour, et années, étant des ajouts triviaux) :

Tuteur1 s’inscrit

Tuteur2 s’inscrit

Edouardo (extérieur à l’université gustave eiffel) dépose 50 euros sur son compte.

Issam (extérieur à l’université gustave eiffel) dépose 50 dollars sur son compte.

Regard des rendez-vous du tuteur1 (aucun)

Edouardo prend rendez-vous avec le tuteur1 lundi matin (accepté, débité)

Issam prend rendez-vous avec le tuteur1 lundi matin (refuser, non débité)

Regard des rendez-vous du tuteur1 (Edouardo)

Tuteur1 et Edouardo font le rendez-vous

Issam reprend rendez-vous lundi matin (accepté, débité)

Regard des rendez-vous du tuteur1 (Issam)

Etudiant1 (de l’université gustave eiffel) recherche des tuteurs avec différents filtres

Etudiant1 prend rendez-vous avec tuteur1 lundi matin

Etudiant1 est mis en attente (rendez-vous avec Issam déjà programmer)

Regard des rendez-vous de l’étudiant1 (aucun)

Tuteur1 et Issam font le rendez-vous

Notification pour étudiant1, son rendez-vous est programmé

Regard des rendez-vous de l’étudiant1 (tuteur1)

Etudiant2 prend rendez-vous avec tuteur1 lundi matin

Etudiant2 est mis en attente (rendez-vous avec Etudiant1 déjà programmer)

Regard des rendez-vous de l’étudiant2 (aucun)

Tuteur1 et Etudiant1 font le rendez-vous

Notification pour étudiant2, son rendez-vous est programmé

Regard des rendez-vous de l’étudiant2 (Tuteur1)

Etudiant3 recherche un tuteur qui fait du python

Etudiant3 en trouve un, il prend rendez-vous

Regard des rendez-vous de l’étudiant2 (Tuteur1)

Regard des rendez-vous de l’étudiant3 (Tuteur2)

Tuteur1 et Etudiant2 font le rendez-vous

Tuteur2 et Etudiant3 font le rendez-vous

Ce scénario démontre la capacité de l'application à gérer la recherche de tuteurs avec des filtres variés, la coordination de prises de rendez-vous simultanées par le service web et le service RMI, la gestion de la mise en attente des rendez-vous avec l'ajout d'un système de notification. Il met également en lumière la fluidité des interactions entre les clients et les tuteurs, ainsi qu'avec l'application elle-même, notamment à travers la consultation des rendez-vous. Enfin, le scénario permet d’illustrer la gestion des paiements avec différentes devises.