

SU – Master 2 – UE DEVREP

JEE – Projet : application de gestion d'inscriptions à une conférence

L'objectif de ce projet est de développer une application des gestions des inscriptions à une conférence, en utilisant les technologies JEE et/ou Spring, ainsi qu'une architecture de microservices. Ce projet est à faire en binôme.

CONSIGNES :

La présente version réduit le périmètre du projet sur les aspects suivants :

- L'application devra juste gérer une conférence à la fois et ne plus proposer de mécanisme pour créer des formulaires d'inscription pour plusieurs conférences actives simultanément. Pour gérer une nouvelle conférence, on doit pouvoir lancer une nouvelle instance de l'application et la configurer.
- L'architecture basée sur les microservices est optionnelle, mais bienvenue. Une architecture classique multi-tiers (couches présentation, métier, données) peut être utilisée pour ce projet. Cependant, il est souhaitable que le back end (couches métier et données) soit packagé de sorte à fournir une API REST au front end.
- Il est souhaitable que votre application soit déployée dans des machines virtuelles (ex : [Vagrant](#)) ou conteneurs, type [Docker](#). Vous pouvez déployer soit toute l'application dans un seul conteneur, soit déployer la base de données dans un conteneur différent du reste de l'application, soit déployer la base de données et le back end (qui fournit l'API REST) dans un conteneur séparé du reste.
- Vous pouvez utiliser les frameworks ou stacks de développement que vous souhaitez, côté front end comme back end, pourvu qu'ils soient modernes et à jour, et que l'application repose sur JEE ou Spring. Par exemple, voici une liste de frameworks de présentation : [Angular](#), [React](#), [Vue](#), [Ember](#), [Backbone](#). Si vous souhaitez faire cohabiter plusieurs frameworks de présentation, [Single SPA](#) ou [iframe](#) seraient utiles.

Vocabulaire :

- Un participant est une personne qui souhaite participer à une conférence et qui remplit un formulaire d'inscription, puis paie son inscription si elle est validée par un administrateur du système.
- Un administrateur est un compte possédant un rôle de super administration de l'ensemble du système ou d'administration de la conférence à laquelle souhaite s'inscrire un participant.

Le flux de travail de l'application côté participant est le suivant :

Demande d'inscription

- Un participant accède au formulaire d'inscription. Les détails du formulaire seront indiqués plus bas dans ce document.
- Il renseigne les champs obligatoires, sélectionne un tarif parmi ceux proposés et fournit une preuve de son statut quand il a choisi un tarif étudiant.
- Il valide ensuite le formulaire.

- Un mail de notification confirmant la demande d'inscription est généré et envoyé au participant.
- Un mail de notification est généré et envoyé aux administrateurs avec les détails de la demande d'inscription.

Païement d'inscription

- Le participant reçoit un mail de confirmation après validation de sa demande d'inscription par un admin. Le mail contient une URL pour aller payer l'inscription.
- Le paiement d'inscription est soumis à un délai d'expiration. Si le participant n'a pas utilisé le lien de paiement avant la fin du délai d'expiration sa demande d'inscription est automatiquement annulée par le système, historisée et une notification lui est envoyée à cet effet. Les administrateurs en sont également notifiés. Le participant doit alors recommencer la procédure par une demande d'inscription.
- Lorsque le paiement est effectué, le participant reçoit une confirmation de paiement avec le reçu de paiement. Les administrateurs reçoivent une notification confirmant ce paiement.
- Lorsque le paiement est effectué, le service de paiement communique l'information sur la transaction au service de gestion des inscriptions (voir plus bas la section Architecture de l'application).

Le flux de travail de l'application côté administrateur est le suivant :

Revue d'inscription

- Un des administrateurs, sur notification de demande d'inscription, va vérifier que la demande est correctement formulée (les champs correctement remplis, attestation d'étudiant fournie quand le participant sélectionne un tarif étudiant).
- Si tout va bien, l'administrateur valide la demande, ce qui génère un mail au participant lui demandant de procéder au paiement de l'inscription, avec l'URL vers le service de paiement.
- La validation déclenche un temporisateur paramétrable de quelques heures par ex., 24h, 48h, 72h, ... La durée est spécifiée par l'administrateur pour cette conférence dont on a ouvert les inscriptions. Si le client n'a toujours pas payé à l'expiration du temporisateur, sa demande est annulée, historisée (pour les statistiques) puis un email de notification lui est envoyé (ainsi qu'aux administrateurs).
- Si quelque chose n'est pas correcte dans la fiche de demande, l'admin rédige un avis de rejet au participant que le système lui envoie (au participant) par email. La fiche est ensuite purement et simplement supprimée. Le participant doit alors recommencer la demande d'inscription.

Païement d'une inscription

- Lorsqu'une inscription a été payée, l'administrateur en est notifié et la fiche d'inscription du participant est mise à jour avec le numéro et la date de la transaction bancaire, le type d'inscription choisie ainsi que le montant payé.
- L'administrateur fait une revue de cette information, puis déclenche (via un menu) la génération du reçu ou facture d'inscription du participant (document PDF) qui est ensuite envoyé à ce dernier. Ce document comporte des champs selon le modèle joint à cet énoncé (cas de Dr Parra – Document réel et confidentiel).
- Le reçu d'inscription est différent du reçu de paiement de l'inscription effectué via le service de paiement (généralement fourni par une banque partenaire).

Exceptions

Il existe des exceptions à la procédure normale des inscriptions comme décrite jusqu'ici.

- Par exemple, les inscriptions des conférenciers invités à délivrer une présentation spéciale (keynotes) offertes (cas du reçu d'inscription du Dr. Engelmann parmi les documents annexes joints à cet énoncé)¹.
- Les inscriptions des membres du comité d'organisation de la conférence sont également offertes (cas de Ms Michelon – étudiante volontaire - et de M. Ziadi – general chair - parmi les documents annexes).
- Les inscriptions de participants provenant d'entreprises ayant sponsorisé la conférence peuvent également être offertes (cas de M. Kuitert parmi les documents annexes).
- Certaines personnalités reconnues dans la communauté des chercheurs organisant la conférence bénéficient d'une inscription à tarif réduit (cas du Prof Appel parmi les documents annexes).

Il faudrait donc définir dans l'application une procédure qui permet de gérer ces cas particuliers.

Création d'une procédure d'inscription pour une conférence

- Le système peut gérer plusieurs conférences simultanément. Toute conférence possède deux périodes d'inscriptions : le **Early Registration** et le **Late Registration**.
- Le montant des tarifs est fixé selon Early Registration et Late Registration, d'où les deux périodes d'inscription pour la même conférence. La période pour les Early registration est suivie le lendemain de sa clôture par la période des Late Registration. Les tarifs Early sont donc plus avantageux que les Late.
- Un formulaire de création d'une procédure d'inscription pour conférence doit permettre à l'administrateur de renseigner l'information essentielle. Tout est en anglais.
 - Nom de la conférence
 - Les champs à inclure dans le formulaire d'inscription, choisis parmi une liste de champs pré-définis :
 - Title (Prof, Dr, Mr, Mrs, Ms)
 - First name
 - Last name
 - Institution
 - Address
 - Zip
 - City
 - Country
 - Email
 - Phone
 - Type d'inscription

Le type d'inscription est une information propre à chaque conférence. Un type d'inscription est composé d'un libellé et d'un tarif (en euros). Le système doit donc permettre à l'administrateur de spécifier une liste de types d'inscription à rattacher au champ *Type d'inscription*. Un exemple de liste

¹ Tous ces documents qui sont des exemples de reçus d'inscriptions sont réels et confidentiels.

déroulante pour le champ *Type d'inscription*, tiré de la conférence SPLC 2019, est donné ci-après.

--Choose registration type--

- Workshop / Tutorial 1-Day Pass (Monday or Tuesday) : 180 €
- Workshop / Tutorial 2-Day Pass (Monday and Tuesday) : 280 €
- Additional Guest - Social Event : 120 €
- Main Conference (Wednesday to Friday) : 550 €
- ✓ Main Conference (Student with paper) : 380 €
- Industry Day : 200 €
- Student without paper : 400 €
- Main Conference + 1 Workshop / Tutorial : 650 €
- Main Conference + 2 Workshops / Tutorial : 700 €

Le tableau général comportant les types d'inscription pour les périodes Early et Late Registrations est donné à titre d'exemple ci-après.

SPLC 2019		
Catégorie d'inscription	Early Registration (du 01/04 au 02/07)	Late Registration (du 03/07 au 23/08)
Workshop / Tutorial 1/2-Day Pass (Monday or Tuesday)	100 €	150 €
Workshop / Tutorial 1-Day Pass (Monday or Tuesday)	180 €	280 €
Workshop / Tutorial 2-Day Pass (Monday and Tuesday)	280 €	400 €
Additional Guest - Social Event	120 €	120 €
Main Conference (Wednesday to Friday)	550 €	700 €
Industrial day	200 €	300 €
Main Conference - Student without paper	400 €	550 €
Main Conference + 1 Workshop / Tutorial	650 €	750 €
Main Conference + 2 Workshops / Tutorials	750 €	850 €

- Le système doit basculer automatiquement les valeurs du champ Type d'inscription du Early vers le Late Registration à la date spécifiée par l'administrateur pour l'application des tarifs du Late Registration.
- Chaque formulaire d'inscription pour une conférence donnée doit être publiquement accessible via une URL spécifique, par ex : <https://conferences.sorbonne-university.fr/splc2019/>
- L'URL général d'accès du système, par ex. : <https://conferences.sorbonne-universite.fr/> ne doit être accessible que sur login/mot de passe aux administrateurs. Les conférences disposent en général de leur propre site indépendant d'information sur leur organisation tandis que l'URL d'inscription est particulière car gérée par un service central de l'université.

Gestion des administrateurs

- Il existe un rôle super administrateur de l'ensemble du système, qu'on peut octroyer à plusieurs comptes utilisateur. Ce rôle possède tous les droits possibles pour administrer l'ensemble du système (y compris créer des comptes).
- Il existe un rôle administrateur de conférences, auquel on peut affecter la gestion d'une ou plusieurs conférences. On peut octroyer ce rôle à plusieurs comptes utilisateur, en désignant, pour chaque compte, les conférences qu'il peut administrer.

Architecture de l'application

Il serait bien de concevoir et développer le système selon une architecture de style microservices. Nous envisageons au moins deux services distincts, qui communiquent :

- Le service principal qui gère les workflows de demande d'inscription, validation, création des conférences, gestion des comptes administrateurs.
- Le service qui fait office de banque pour simuler le paiement d'une demande inscription validée.

Vous pouvez rescinder le service principal en plusieurs services distincts, par exemple :

- un service qui gère les demandes d'inscription, leur validation et la confirmation des paiement, ainsi que les historiques et l'affichage de toutes ces données (demandes en cours, validées, rejetées, annulées, paiements en attente, confirmés, qui se sont soldés par un échec) ;
- un service qui gère l'administration du système (comptes, création et configuration de procédure d'inscription à des conférences).

Logging

Pour logger² les événements pendant le fonctionnement du système, utilisez la bibliothèque Logback (<http://logback.qos.ch/index.html>), qui est une implémentation de la spécification slf4j (<http://www.slf4j.org>). Dans le cas d'un projet Maven, le fichier de configuration logback.xml se place dans src/main/resources/.

Vous pouvez aussi utiliser toute bibliothèque de logging standard que votre framework de programmation de l'application vous recommandera d'utiliser.

Restitution de ce projet

Les artefacts du projet à rendre :

- code source avec toute la configuration de déploiement de l'application, base de données, les mots de connexion à l'application ;
- documentation utilisateur final (i.e. les participants qui font une demande d'inscription et ceux qui finissent par s'inscrire) ;
- documentation administrateur avec la procédure pour déploiement, démarrage et connexion à l'application (déploiement sous Docker ou Vagrant souhaitable).

² <https://stackify.com/logging-logback/>

Rendez tous les artefacts créés lors de ce projet une semaine après la dernière séance de DevRep, soit à travers le dépôt de code source que vous aurez créé (ex., Github, Gitlab, BitBucket), soit par une archive .ZIP téléchargeable d'un drive en ligne.