

Master DL – 2014-2015 – Projet Java EE

1. Introduction

Le projet Java EE vise la mise en pratique des notions abordées en cours et en TP dans le cadre d'une production à réaliser en binôme.

Chaque groupe a pour mission le développement de l'application Web « ToulouseMusee ».

La mairie de Toulouse souhaite faciliter l'accès aux musées pour les touristes présents sur la ville de Toulouse sur une période donnée. Elle vous confie donc le développement de l'application ToulouseMusee qui permet à un utilisateur souhaitant effectuer une ou plusieurs visites guidées dans les musées de Toulouse d'enregistrer sa demande via Internet.

Les informations sur les musées de Toulouse sont contenues dans le fichier intitulé « Musee.csv » (le fichier Musee.csv provient de l'initiative Open Data de la ville de Toulouse <https://data.toulouse-metropole.fr/>).

2. Exigences fonctionnelles - « User stories »

2.1. Recherche de Musées

Description

En tant qu'utilisateur,
Je peux effectuer une recherche de Musées,
En vue d'effectuer une demande de visites.

Tests d'acceptations

Étant donné :

- l'URL d'accès à l'application

Quand :

- J'accède à la page d'accueil de l'application depuis mon navigateur

Alors :

- je dispose de l'interface du moteur de recherche de Musée ;
- les critères de recherche disponibles dans l'interface sont les suivants :
 - chaîne de caractères contenue dans le nom du musée
 - Code postal (liste déroulante)
 - chaîne de caractères contenue dans le nom de la rue du musée
- aucun musée n'apparaît sur la page

Quand :

- j'effectue une recherche de musée selon les critères disponibles dans l'interface de recherche

Alors :

- La liste de résultats affiche pour chaque musée correspondant à la recherche :
 - son nom
 - son numéro de téléphone
 - son adresse
 - ses accès métro et bus
 - les horaires
 - son gestionnaire
- les résultats s'affichent avec un maximum de 5 musées par page.

2.2. Sélection d'un musée pour ajout à la liste de « mes musées préférés»

Description

En tant qu'utilisateur navigant sur les pages affichant les résultats d'une recherche, Je peux ajouter un musée à la liste des musées qui m'intéressent En vue de spécifier une demande de visite pour ces musées.

Tests d'acceptations

Étant donné :

- la page d'accueil de l'application affichant une liste de musées contenant au moins 3 musées

Alors :

- chaque ligne de résultat correspondant à un musée fait apparaître un bouton « ajouter à ma liste de musées » .

Quand :

- je clique sur le bouton « ajouter à ma liste de musées » correspondant à un musée

Alors :

- la liste intitulée « mes musées préférés » apparaît dans un encart à droite de l'écran,
- la liste affiche le nom du musée que je viens de sélectionner,
- je ne peux plus cliquer sur le bouton me permettant d'ajouter ce même musée à la liste

Quand :

- je clique sur le bouton « ajouter à ma liste de musées » correspondant à un autre musée

Alors :

- la liste de mes musées préférés est mise à jour et affiche donc les noms des deux musées sélectionnés triés par ordre croissant sur les noms.

2.3. Suppression d'un musée de la liste « mes musées préférés »

Description

En tant qu'utilisateur,
Je peux supprimer un musée de la liste des musées qui m'intéressent,

Tests d'acceptations

Étant donné :

- la page d'accueil de l'application affichant dans la liste « mes musées préférés » au moins 2 musées

Alors :

- chaque ligne correspondant à un musée dans la liste « mes musées préférés » fait apparaître un bouton « supprimer de ma liste de musées » .

Quand :

- je clique sur le bouton « supprimer de ma liste de musées » correspondant à un musée

Alors :

- le musée correspondant est supprimé effectivement de la liste « mes musées préférés »

Quand :

- j'effectue une recherche qui fait apparaître le musée que je viens de supprimer de ma liste de musées préférés

Alors :

- le bouton «ajouter à ma liste de musées » est de nouveau actif pour ce musée.

2.4. Demande de visite

Description

En tant qu'utilisateur ayant sélectionné des musées dans ma liste de musées préférés,
Je peux effectuer une demande de visite pour ces musées pour un maximum de 6 personnes,
En vue de visiter ces musées en famille.

Tests d'acceptations

Étant donné :

- la page d'accueil de l'application affichant dans la liste « mes musées préférés » au moins 1 musée

Alors :

- je dispose d'un bouton dans l'encart « mes musées préférés » intitulé « Effectuer une demande de visite ».

Quand :

- je clique sur le bouton « Effectuer une demande de visite »

Alors :

- je suis envoyé sur une nouvelle page pour la saisie de demande de visites,
- la page affiche la liste de mes musées préférés
- la page affiche le formulaire de saisie contenant les champs suivants
 - la date de début et de fin de la période pendant laquelle je souhaite réaliser la visite,
 - le nombre de personnes présentes à la visite avec une indication que le nombre de personnes ne peut excéder 6.

Quand :

- je saisis dans les champs de formulaire des informations non valides ET
- je soumetts le formulaire

Alors :

- un message d'erreur s'affiche sur la page
- je peux de nouveau saisir les informations dans les champs de formulaire.

Quand :

- je saisis les informations demandées valides ET
- je soumetts le formulaire

Alors :

- l'application me communique mon code de demande accompagné d'un message indiquant que la demande sera traitée prochainement.

2.5. Accès à mes demandes de visite (Optionnel)

Description

En tant qu'utilisateur,
Je dispose d'un accès à mes demandes,
de telle sorte que je puisse contrôler l'état d'avancement du traitement de mes demandes.

Tests d'acceptations

Étant donné :

- un utilisateur accédant à la page d'accueil de l'application

Alors :

- je dispose d'un lien dans une barre de menu horizontale en haut de la page me permettant d'accéder à la liste de mes demandes

Étant donné :

- un utilisateur ayant au moins une demande en cours accédant à la page d'accueil de l'application

Quand :

- je clique sur le lien d'accès à mes demandes

Alors :

- je suis envoyé sur une nouvelle page,
- un formulaire me permet de saisir le code de la demande.

Quand :

- je saisis un code erroné et
- je soumetts le formulaire

Alors :

- un message d'erreur s'affiche au dessus du formulaire de saisie du numéro de demande,
- je peux saisir un nouveau code.

Quand :

- je saisis un code correspondant à une de mes demandes

Alors :

- la demande s'affiche en récapitulant toutes les informations de la demande ainsi que son statut (En cours de traitement, Confirmée, Refusée)
 - si la demande est refusée, le message d'explication suivant doit accompagner le statut : « Aucun guide disponible sur cette période »,
 - si la demande est acceptée, la date de début de la visite et sa date de fin doivent être indiquées.

3. Exigences non fonctionnelles

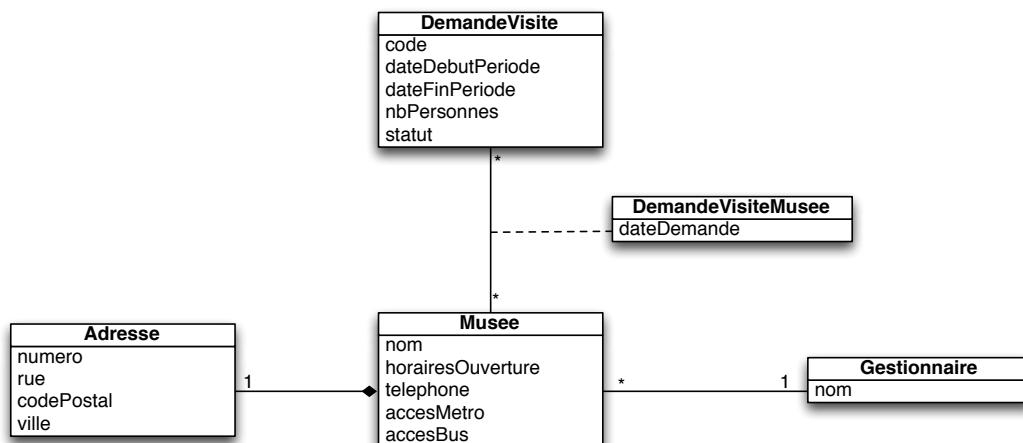
3.1. Environnement technique

Framework de développement	Grails 2.4.2
IDE	IntelliJ version 13.1 .1 ou supérieure
Framework de test	Spock (version packagée Grails)
SGBDR	H2 (version packagée Grails)
Mesure de la couverture de test	Plugin coverage 2.0.3-3 (cobertura)
Gestion de version de code	Git
Repository Git partagé	GitHub*

* Chaque étudiant devra donc créer si nécessaire un compte GitHub ; un projet Open Source devra être créé sur un des comptes GitHub du binôme pour héberger le code source du projet. Les commits respectifs de chaque contributeur permettront d'évaluer la participation effective de chacun membre du binôme.

3.2. Accès à la base de données

- Le modèle de données doit implémenter le diagramme de classes suivant en tenant compte des problèmes que peut poser la modélisation explicite des relations to-many dans Gorm/Hibernate :



- Les accès à la base en écriture doivent s'exécuter de manière transactionnelle.
- Les accès à la base ne doivent générer aucun problème de « N+1 select ».
- La stratégie de gestion des accès concurrents en écriture s'effectue en mode « optimistic locking » utilisant un « Timestamp » comme attribut modifié à chaque mise à jour (Optionnel).

3.3. Qualité

- Les classes du modèle doivent faire l'objet de tests conçus avec Spock avec une couverture minimale de 80% en nombre de lignes.

- La mesure de la couverture de test s'effectuera à l'aide du plugin « coverage ».
- Le jeu de tests utilisera la liste de tous les musées contenus dans le fichier « Musee.csv ».
- Le jeu de test servira aux tests Spock ainsi qu'aux tests fonctionnels manuels : pour cela l'initialisation du jeu de test devra être invoquée dans la closure « init » de la classe Bootstrap.

3.4. Livrables du projet

Item	Format	Livraison sur	Date
Code source du projet Grails	-	Github 1 tag créé pour la livraison	1 jour avant restitution
Formulaire projet complété	Google form*		
Support de la restitution orale	PDF	Moodle.	1 jour avant la restitution

* https://docs.google.com/forms/d/1hVzeub8zph4sRObWkA06VD4B0eR-96O9okVM9-zLPio/viewform?usp=send_form

3.5. Restitution du projet

La restitution du projet s'effectuera sous la forme d'une présentation orale (date fixée dans l'emploi du temps).

Les membres du binôme présenteront leur réalisation en mettant l'accent sur les moyens techniques mis en œuvre pour répondre à l'ensemble des exigences fonctionnelles et non fonctionnelles.