Rapport de laboratoire

**Département de génie logiciel et des technologies de l’information**

|  |  |
| --- | --- |
| **No de laboratoire** | 3 |
| **Étudiant(s)** | François Gagné  Patrick Lavallée  Julien Pelletier-Morin  Simon Turcotte |
| **Code(s) permanent(s)** | GAGF20067801  LAVP12048408  PELJ03098909  TURS17118905 |
| **Cours** | GTI525 |
| **Session** | Hiver 2013 |
| **Groupe** | 01 |
| **Professeur(e)** | Éric Boivin |
| **Chargé(e) de laboratoire** | Mohamed Outellou |
| **Date** | CHANGE THE DATE !!! |

Table des matières

[1 Introduction 3](#_Toc352017432)

[2 Conception 4](#_Toc352017433)

[2.1 Diagramme d’entité-relation 4](#_Toc352017434)

[2.2 Diagramme de classes 4](#_Toc352017435)

[3 Discussion 5](#_Toc352017436)

[4 Conclusion 6](#_Toc352017444)

# Introduction

Dans le cadre de ce cours, l’équipe doit concevoir et déployer un système de vente de billets en ligne. Il s’agit de la deuxième itération du projet qui vise à dynamiser l’expérience utilisateur notamment en ajoutant une couche applicative qui répondra à des requêtes HTTP.

Lors de la première itération, un prototype statique a été élaboré. Ce prototype est dynamisé en transformant les pages statiques au format JSP et en incluant des mécanismes d’obtention de données se trouvant sur un serveur distant. Ces données sont pour l’instant « *stubbées »*. De plus, la plateforme *Spring* est mise de l’avant et favorise une approche MVC qui est un standard de développement Web.

De plus, le diagramme de classe est mis à jour et est une évolution des concepts élaborés précédemment. Finalement, une discussion est fournie et démontre les difficultés et les bons coups de l’équipe.

# Conception

On est pas jambon pis on fournis les deux. Le premier va de soi pis le deuxieme c’est par rapport à nos classes hibernate qu’on a rajouté. That’s it.

## Diagramme d’entité-relation

## Diagramme de classes

# Discussion

## Quels sont les patrons de persistance qui ont été utilisés dans votre application? Si vous avez utilisé un framework, quels patrons ont-ils permis d’implémenter?

## Quel niveau d’effort a été nécessaire afin de passer d’un modèle basé sur les « stubs » au modèle avec persistance?

## Quelles sont les principales failles de sécurité de votre application?

## Quels moyens ont été mis en place afin d’éviter les failles de sécurité?

## Avez-vous utilisé une technologie où un framework additionnel? Si oui, expliquez quel est le rôle de cette technologie dans votre application.

## Quelles modifications feriez-vous au service Web de paiement par carte de crédit?

## Quelle a été, selon vous, l’élément le plus complexe à réaliser durant la session?

# Conclusion

Le but de cette deuxième itération consistait en la dynamisation de la solution logicielle déployée lors du laboratoire précédent. L’utilisation de la technologie JSP a permis à l’équipe d’insérer de la logique Java à l’intérieur d’un document HTML.

Suite à une contrainte imposée due au nombre de membres de l’équipe, la plateforme *Spring* a été la technologie centrale du développement du livrable de cette itération. Qui plus est, l’équipe a su tirer avantage d’inclure des tests automatisés tôt dans le cycle de développement permettant d’effectuer des tests de régression.

Le résultat est une application web permettant de consulter du contenu obtenu dynamiquement. Elle envoie des requêtes HTTP vers un serveur qui traite la requête, obtient les données, applique la logique d’affaires et retourne un résultat interprétable par un navigateur.

Finalement, l’utilisation d’un *« Stub »* pour simuler la persistance des données a permis de bien découpler les classes logicielles impliquées pour permettre à l’application de migrer vers l’utilisation de la plateforme *Hibernate* lors de la prochaine itération.