Hashcode

**Groupe No :** 6

Rapport de construction et de transition

Butty Joé

Fuchs Nicolas

Rial Jonathan

**Filière :** Informatique

**Technologies :** Java EE, MySQL

**Date de rendu :** 20.05.2018

**Superviseurs :** Prof. Houda Chabbi Drissi

Prof. Pierre Kuonen

Prof. Omar Abou Khaled

**Client:** Prof. Pierre Kuonen

**Table des matières**

1 Implémentation SGBD 4

1.1 Vos choix en tant qu’administrateur de votre SGBD 4

1.1.1 Estimation de la taille des données 4

1.1.2 Comptes sur votre SGBD 4

1.2 Vos choix pour l’optimisation et la concurrence/synchronisation 4

1.2.1 Choix d’index pour chaque relation 4

1.2.2 Argumentation sur la dénormalisation 4

1.2.3 Choix pour la concurrence 4

1.2.4 Les procédures stockées et vues 5

1.3 Etude d’un plan d’exécution 5

1.4 Code SQL dans cet ordre 5

1.4.1 Code de l’implémentation des tables 5

1.4.2 Code des triggers 5

1.4.3 Code des procédures stockées et des fonctions implémentés 5

1.4.4 Code de création des utilisateurs avec leurs privilèges 5

1.4.5 Remarques 5

1.5 Caractéristiques SGBD 5

1.5.1 Explicitez les triggers et leurs caractéristiques 5

1.5.2 Explicitez les niveaux d’isolations de votre SGBD et le protocole qu’il utilise 5

1.5.3 Le support de XML et XSD 5

1.5.4 Les types d’index disponibles 5

2 Implémentation SI 6

2.1 Environnement spécifique de développement 6

2.1.1 Framework 6

2.2 Éléments architecturaux 6

2.2.1 Couche présentation 6

2.2.2 Couche métier 7

2.2.3 Couche service 7

2.3 Éléments technologiques 8

2.3.1 Outils de gestion de versionning (CVS / SVN) 8

2.3.2 Outils de modélisation 8

2.3.3 Outils de documentation du code 8

2.3.4 Outils de journalisation de prototype 8

2.3.5 Outils de test 9

2.3.6 Outils de gestion de licences 9

2.3.7 Outils de d’installation et de packaging 9

2.4 Diagrammes d’architecture 10

2.4.1 Diagramme(s) de package 10

2.4.2 Diagramme(s) d’état-transition 10

2.4.3 Diagramme de déploiement 11

2.5 Choix technologique 12

2.6 Contraintes d’utilisation technologique 13

2.7 Outils administratifs 14

3 Conclusion 14

3.1 Objectifs atteints 14

3.2 Problèmes rencontrés 15

3.3 Problèmes non résolus 15

3.4 Perspectives futures 15

3.5 Synthèse 15

4 Références 16

4.1 Bibliographie 16

4.1.1 Thème : Reconnaissance vocale 16

4.2 Webographie 16

4.2.1 Thème : Reconnaissance vocale 16

4.3 Projet 16

4.3.1 Thème : Reconnaissance vocale 16

5 Annexes 17

5.1 Support numérique (une copie par rapport) 17

# Implémentation SGBD

## Vos choix en tant qu’administrateur de votre SGBD

### Estimation de la taille des données

Cf. chapitre organisation physique

Estimation moyenne :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table** | **NbTuples** | **Nbchamps** | **Formule Calcul** | **Taille tuple** | **Taille Table** |
| role | 5 | 2 | 5\*4 +6 + 20 + 18 + 15 + 13 | 10/24/22/19/17 | 92 |
| account | A | 9 |  |  |  |
| challenge | B | 7 |  |  |  |
| team | C | 4 |  |  |  |
| account\_team | D | 2 | D\*8 | 8 | D\*8 |
| challenge\_organizer | E | 2 | E\*8 | 8 | E\*8 |
| solution | F | 9 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Taille des champs utilisés :

|  |  |
| --- | --- |
| **Type** | **Taille** |
| int | 4 bytes |
| float | 4 bytes |
| datetime | 8 bytes |

Justifiez alors de la taille de la BD (n’oubliez pas les index) que devez réserver à sa création en tenant compte de l’évolution de la taille de ces données suivant votre application.

### Comptes sur votre SGBD

Choix pour implémenter la sécurité (revoke et grant) : Lien entre vos vues externes (rapport analyse) et les comptes crées.

## Vos choix pour l’optimisation et la concurrence/synchronisation

### Choix d’index pour chaque relation

* Commencer par présenter les choix que vous avez fait pour l’implémentation des relations et des index à leur associer.
* Prenez une de vos relations et montrer l’apport de l’ajout d’un index en étudiant les temps d’exécution d’une requête que vous aurez choisi.

### Argumentation sur la dénormalisation

Étude des performances et choix de dénormalisation (duplication de l'information, fusion de tables...), choix relatifs aux données dérivées ajout de clés numériques, clustérisation…..

### Choix pour la concurrence

Décrire clairement les ajouts de tables/colonnes, …. Que vous avez dû faire pour assurer la concurrence.

### Les procédures stockées et vues

Les procédures stockées et vues mises en places pour optimiser les jointures etc. et géré la sécurité…..

## Etude d’un plan d’exécution

Isoler une requête compliquée de votre application, ou créez-en une à la main.

1. Présenter la requête (SQL),
2. Votre plan d’exécution en AR
3. Le plan d’exécution généré par le SGBD
4. Commentez clairement ce qui se passe.

## Code SQL dans cet ordre

Attention aux CIs indiquées dans le modèle relationnel : il doit y avoir un commentaire qui indique l’endroit où elles ont été implémentées. **Utiliser leur nom.**

### Code de l’implémentation des tables

### Code des triggers

### Code des procédures stockées et des fonctions implémentés

### Code de création des utilisateurs avec leurs privilèges

### Remarques

Mettre ici toute astuce qui a dû être utilisée parce que le SGBD utilisé ne permet pas une gestion de quelque chose que vous pensiez utiliser : gestion des exceptions etc.

## Caractéristiques SGBD

Complétez les sections suivantes qui permettent de comprendre le SGBD utilisé et ses caractéristiques.

* NOM du SGBD : MySQL
* Version :
* Références d’où proviennent vos commentaires :

### Explicitez les triggers et leurs caractéristiques

### Explicitez les niveaux d’isolations de votre SGBD et le protocole qu’il utilise

### Le support de XML et XSD

### Les types d’index disponibles

# Implémentation SI

## Environnement spécifique de développement

### Framework

Il s’agit de remplir et compléter les informations sur des éventuelles couches et Framework qui ont été rajouté sur l’environnement de base :

*Par exemple :*

1. *Couche JSF*
2. *Couche AJAX*
3. *lus d’infos & liens :*
   1. [*https://glassfish.dev.java.net/*](https://glassfish.dev.java.net/)

## Éléments architecturaux

### Couche présentation

Il s’agit de remplir et compléter les informations suivantes :

1. Type de l’IHM
   1. Client léger.
      1. Pourquoi ? : ….
   2. Client lourd.
      1. Pourquoi ? : ….
   3. Client mixte.
      1. Pourquoi ? : ….
2. Support du IHM
   1. Navigateur (version :….). Pourquoi ?
   2. Autre approche. Laquelle ? Pourquoi ? ….
3. Explication sur le choix technologique :
   1. Explications générales :
      1. Ex. pour la couche présentation, l’approche utilisé est d’implémenter un client riche avec AJAX parce que…
   2. Plateforme :
      1. Ex. Dojo, Pourquoi ?
4. Réalisation de l’interface
   1. L’interface IHM est réalisée « from scratch » par le développeur.
      1. Pourquoi et comment ?
   2. L’interface IHM est réalisée par l’environnement de développement
      1. Pourquoi et comment ?

Nous avons décidé d'utiliser Angular 2 pour faciliter le traitement des données du côté client mais également pour obtenir plus rapidement des interfaces IHM visuellement correctes sans y consacrer trop de temps.

* 1. L’interface IHM est réalisée par le biais d’un plugin ou d’un programme externe.
     1. Lequel ? (ex. Skincrafter)
     2. Pourquoi et comment ?
  2. Plus d’infos & liens :
     1. [*http://www.telerik.com*](http://www.telerik.com/)
     2. [*http://www.skincrafter.com*](http://www.skincrafter.com/)
     3. *https://woodstock.dev.java.net/index.html*

### Couche métier

Il s’agit de remplir et compléter les informations suivantes :

1. Inventaire des servlets et des JSP / ASPX et CS / PhP avec un tableau après le diagramme de package avec une explication succincte de chaque classe. Ex. diagramme de package

|  |  |  |
| --- | --- | --- |
| Nom package | Lite des classe | Explications |
|  |  |  |
|  |  |  |
|  |  |  |

1. Techniques de mapping BD vs. XML schémas : il s’agit de fournir des explications sur les moyens utilisés pour assurer le mapping entre les champs de la base de données et les éléments XML nécessaires à la publication :
   1. Feuilles de style XSLT. il s’agit de présenter le(s) feuille(s) de styles commentées, un exemple d’instance XML et un exemple de résultat(s).
   2. Programmation classique. il s’agit de présenter le(s) programme(s) commentés, un exemple d’instance XML et un exemple de résultat(s).
   3. Autre. il s’agit de présenter le(s) approche(s) commentées, un exemple d’instance XML et un exemple de résultat(s).
2. Les Technologies XML pour publication
   1. From scratch. Pourquoi et comment ?
   2. Des APIs externes. Lesquelles ? Pourquoi et comment ?

### Couche service

Il s’agit de remplir et compléter les informations suivantes :

1. L’accès à la base de données : il s’agit de donner des informations sur le connecteur vers la base de données.
   1. Nom : mysql-connector-java
   2. Version : 6.0.6
   3. Plateforme : multi-plateforme
   4. Configuration : spring.datasource.url=jdbc:mysql://localhost:3306/mydb?useSSL=false
2. La gestion de la politique et des droits d’accès : il s’agit de donner des informations sur les éléments de gestion de la politique d’accès.
   1. Interne à la base de données
      1. Utilisateurs
      2. Rôles et droits
   2. Externe à la base de données
      1. Où ?
         1. Web.xml. Configuration :
         2. Web.config. Configuration :
      2. Autre : … Configuration :
3. La gestion des aspects de sécurité : il s’agit de donner des informations sur les éventuelles techniques de protection de passwd, de cryptographies, du protocole https, etc.

## Éléments technologiques

### Outils de gestion de versionning (CVS / SVN)

Il s’agit de donner les plus amples informations sur le système de gestion de version utilisé dans le cadre de votre projet : L’adresse du serveur, Le client utilisé, Le chemin d’accès, Synthèse sur les avantages et les facilités d’utilisation, Synthèse sur les désavantages et les difficultés d’utilisation.

L'outil de gestion de versionning que nous avons utilisé est git. C'est un outil très complet et efficace mais qui demande aussi un apprentissage en profondeur lorsque la structure du projet se complexifie. Il n'est pas évident non plus de gérer du code avec les fichiers temporaires et de configuration qui changent d'une machine à l'autre. Pour cela, l'utilisation du gitignore est requise.

### Outils de modélisation

Il s’agit de donner les plus amples informations sur l’utilisation de XXXX au niveau :

1. La génération du code (C#, java, ou PHP)
2. La génération des documents de documentation pour développeurs (Doc, Excel, et PowerPoint)
   1. Ex. de chaque Template
3. Plus d’infos & liens : http://staruml.sourceforge.net/en/

### Outils de documentation du code

Il s’agit de donner des informations sur l’outil(s) de génération de documentation du code :

1. Outil natif à l’environnement de développement
   1. Synthèse sur son utilisation
2. Outil externe. Lequel ? ex. javadoc, ndoc
   1. Synthèse sur son utilisation

Javadoc

1. Ex. de commentaire avec syntaxe complet (class, attribut, commentaires,)
2. Ex. de résultat de cette génération sur un fichier code
3. Plus d’infos & liens :
   1. <http://www.winnershtriangle.com/w/>
   2. <http://www.codeproject.com/KB/macros/kingstools.aspx>
   3. <http://sandcastle.codeplex.com/>

### Outils de journalisation de prototype

Il s’agit de donner des informations sur l’outil(s) de journalisation de prototype.

1. Outil natif à l’environnement de développement
   1. Synthèse sur son utilisation
2. Outil externe. Lequel ? ex. log4java
   1. Synthèse sur son utilisation
3. Ex. de résultats commentés
4. Plus d’infos :
   1. <http://logging.apache.org/>
   2. <http://en.wikipedia.org/wiki/Log4j>
   3. http://logging.apache.org/log4net/
   4. <http://lutecefalco.developpez.com/tutoriels/dotnet/log4net/introduction/>
   5. <http://www.dotnetspace.com/articles/general-articles/using-log4net---very-quick-start.html>
   6. http://www.kellermansoftware.com

### Outils de test

Il s’agit de donner des informations sur l’outil(s) de test « Fonctionnelle » et « unitaire ».

1. Outil natif à l’environnement de développement
   1. Synthèse sur son utilisation
2. Outil externe. Lequel ? ex. ncover, junit, rational functional Tester
   1. Synthèse sur son utilisation
3. Ex. de résultats commentés
4. Plus d’infos & liens:
   1. <http://fr.wikipedia.org/wiki/Test_unitaire>
   2. <http://en.wikipedia.org/wiki/Functional_test>
   3. <http://en.wikipedia.org/wiki/Unit_test>
   4. <http://www.geekzone.co.nz/vs2008/4819>
   5. <http://seleniumhq.org>

### Outils de gestion de licences

Il s’agit de donner des informations sur l’outil(s) de gestion de licence.

1. Outil natif à l’environnement de développement
   1. Synthèse sur son utilisation
2. Outil externe. Lequel ? ex. softwarekey
   1. Synthèse sur son utilisation
3. Ex. de résultats commentés
4. Plus d’infos & liens:
   1. [http://www.softwarekey.com](http://www.softwarekey.com/)
   2. <http://www.xheo.com/Products/Licensing>
   3. <http://www.mancosoftware.com/licensing/howto.htm>
   4. <http://www.gnu.org/licenses/gpl.html>
   5. <http://www.infralution.com/licensing.html>
   6. <http://www.jproductivity.com/products/protection/developer.htm>
   7. <http://members.chello.at/stefan-schneider/JavaLicenseKit.html>

### Outils de d’installation et de packaging

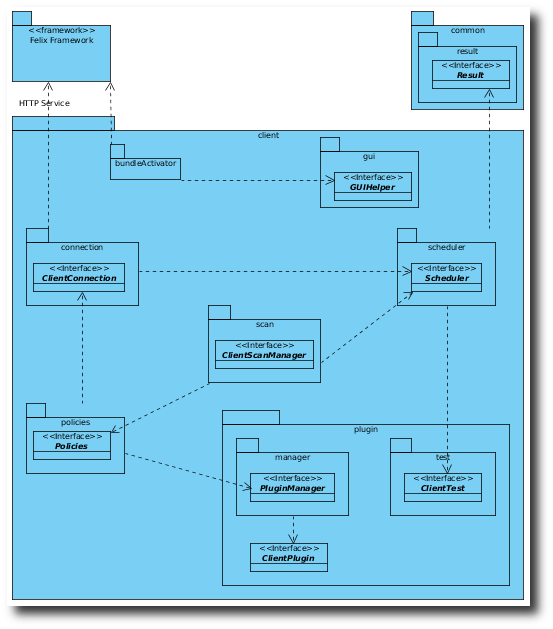
Il s’agit de donner des informations sur l’outil(s) d’installation et de packaging.

1. Outil natif à l’environnement de développement
   1. Synthèse sur son utilisation
2. Outil externe. Lequel ? ex. installshield
   1. Synthèse sur son utilisation
3. Ex. de résultats commentés
4. Plus d’infos & liens:
   1. <http://en.wikipedia.org/wiki/InstallShield>
   2. <http://www.innosetup.com/isinfo.php> (ex. pour ruby et autre)
   3. <http://www.pantaray.com/>

## Diagrammes d’architecture

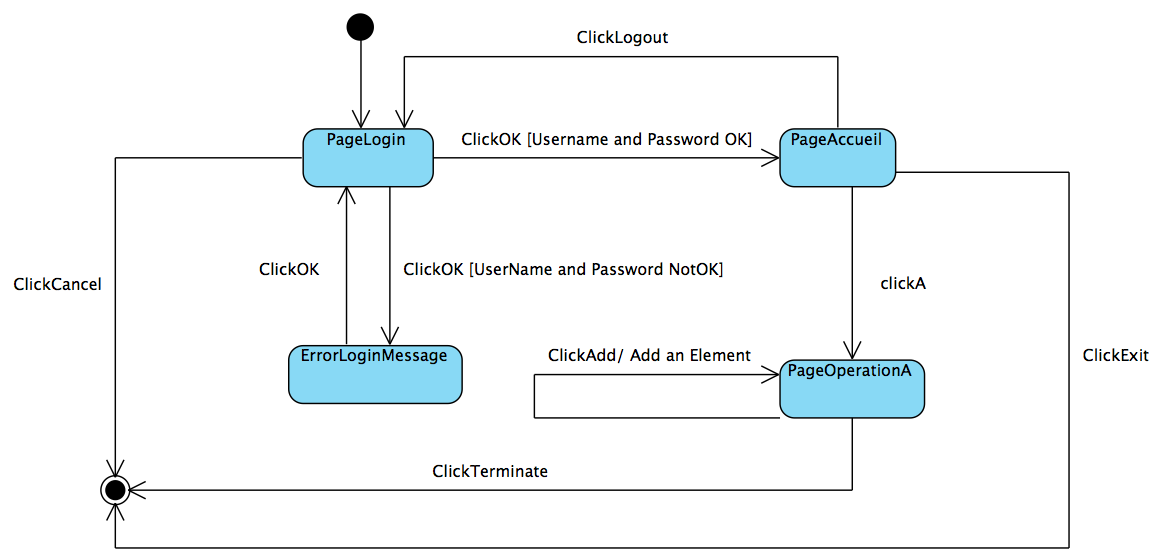
### Diagramme(s) de package

Utilisé pour séparer le modèle en conteneurs logiques, et décrire leurs interactions à un haut niveau. Il s’agit de fournir un diagramme de paquetage au format UML commentée en détails. Ci-dessous un exemple de diagramme de paquetage.

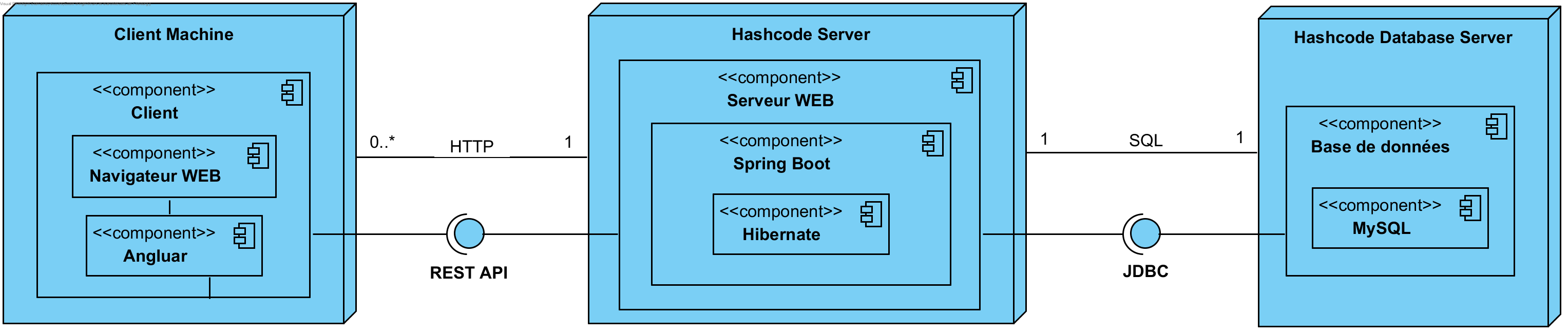


### Diagramme(s) d’état-transition

Il s’agit de schématiser la navigation entre les pages (interface utilisateur) constituant votre application. Pour cela vous allez utiliser un diagramme d'état-transition. Chaque état correspond à une page et les transitions sont les évènements qui font passer d'une page à l'autre (par exemple un « click » de l'utilisateur. Ci-dessous un exemple simple.



### Diagramme de déploiement

Il s’agit de fournir une illustration commentée en détails des éléments formant le diagramme. Ce diagramme est une version complète qui indique principalement quelles machines sont impliquées et pour chaque machine quelle plateforme/outil est déployé. On indiquera également pour chaque lien la technologie utilisée pour la communication ainsi que les multiplicités. Le diagramme représentera le déploiement maximal possible de l'application et pas forcément la façon dont l'application sera effectivement déployée à la fin de projet. Indiquez également quels composants sont déployés sur chaque nœud. Ces composants doivent être cohérents avec ceux du document d'élaboration. S'il y a des changements redonnez le nouveau diagramme de composants dans une figure séparée.

## Choix technologique

A remplir et modifier selon les besoins du projet

|  |  |  |  |
| --- | --- | --- | --- |
| **Approche technologique et langages** | **IHM** | **Logique / Middleware** | **Data** |
| JDBC – préciser la version |  |  |  |
| AJAX – préciser la plateforme ou les librairies utilisées |  |  |  |
| Web Services – préciser les librairies |  |  |  |
| DOM – préciser les librairies |  |  |  |
| SAX– préciser les librairies |  |  |  |
| SQL |  |  |  |
| XML schéma, validation avant stockage dans la base |  |  |  |
| Fichiers de configuration : Java EE (web.xml, etc. ) |  |  |  |
| Fichiers de configuration : XXXX (xxx.xxx, etc.) |  |  |  |
| Browser - iexplorer |  |  |  |
| Browser - firefox |  |  |  |
| Browser - xxxx |  |  |  |

## Contraintes d’utilisation technologique

A remplir et modifier selon les besoins du projet

|  |  |  |
| --- | --- | --- |
|  | OUI (justifier) | NON (justifier) |
| Utilisation une architecture 3 tiers au minimum | La plupart des applications actuelles fonctionnent de cette manière : partie cliente, partie serveur et partie base de données |  |
| Utilisation des web services dans votre architecture | Le serveur mis en place est de type REST. C'est très rapidement implémenté et c'est ce qui est proposé par défaut dans une application Spring Boot. |  |
| Utilisation des RIA dans votre architecture |  |  |
| Utilisation au minimum d’une feuille de style XSLT |  |  |
| Utilisation au minimum d’une feuille de style XSL-FO |  |  |
| Utilisation du SQL | Les requêtes lancées sur notre base de données MySQL utilisent SQL |  |
| Utilisation du XQuery |  |  |
| Utilisation au minimum d’un contrôle de validation sur la saisie des données | MySQL ne supporte aucune validation XML donc cette validation doit être faite au niveau serveur |  |
| Utilisation de DOM, SAX ou JDOM, JAXP |  |  |
|  |  |  |
|  |  |  |

## Outils administratifs

|  |  |  |
| --- | --- | --- |
|  | Oui (justifier) | Non (justifier) |
| Outil de gestion de planning |  | Le planning n'a pas été définit sur papier mais oralement au sein du groupe de travail. |
| Gestion tache, etc. | http://www.clockingit.com/ |  |
|  |  |  |
|  |  |  |

# Conclusion

## Objectifs atteints

La conclusion résume très succinctement ce qui a été réalisé. Cette sous-section se terminera par un tableau de la forme suivante :

|  |  |
| --- | --- |
| Objectifs |  |
| UseCase1 | 100% |
| UseCase2 | 90% |
| …. | 100% |
| Test (functional, unit, etc.) | 100% |
|  |  |

## Problèmes rencontrés

Joé qui emmerde

## Problèmes non résolus

Joé qui emmerde

## Perspectives futures

Joé qui emmerde

Cette sous-section contient des propositions pour des amélioration/évolution futur du produit.

## Synthèse

Cette sous-section contient une courte indication de l’intérêt (ou du non-intérêt) de ce PI sur le plan pédagogique et le point de vue sur l’organisation de ce laboratoire, …

# Références

A remplir et modifier selon les besoins du projet

## Bibliographie

### Ex. Thème : Reconnaissance vocale

[B1], SALT Specification 1.0. 15 juillet 2002, Copyright Cisco Systems, Intel, Microsoft, Philips, SpeechWorks

**Votre conseil relativement à cette référence:** bonne référence pour résoudre le problème xxx

[B2], VoiceXML Specification 1.00. 07 mars 2000, Copyright VoiceXML Forum.

**Votre conseil relativement à cette référence:** à éviter les explicitons non claire

## Webographie

### Ex. Thème : Reconnaissance vocale

[S1], SALT: http://www.saltforum.org

**Votre conseil relativement à cette référence**: bonne référence pour résoudre le problème xxx

[S2], VoiceXML: http://www.voicexml.org

**Votre conseil relativement à cette référence:** bonne référence pour résoudre le problème xxx

## Projet

### Ex. Thème : Reconnaissance vocale

[P1], DynaVoice, Cédric Delléa. 2003, Projet de travaile de diplôme HES à l’HEVs

**Votre conseil relativement à cette référence**: bonne référence pour résoudre le problème xxx

[P2], Etude d’applications VoiceXML dans le cadre de PME avec des petites succursales, Christian Bardet, Sacao SA 2003, Projet de travail de diplôme EduSwiss

**Votre conseil relativement à cette référence**: bonne référence pour résoudre le problème xxx

# Annexes

## Support numérique (une copie par rapport)

[1]répertoire 1 : Répertoire de travail du PI

* Rapport, présentation, etc., Procès verbaux, les emails, etc.

[2]répertoire 2 : Développement logiciels

* Sources, Librairies (spécifique, communes), Serveurs ou autres produits.