|  |
| --- |
| Biere Lab  Documentez votre application communautaire  Dossier de conception technique  Version 1.0 |
| **Auteur**  Delomez Matthieu  Développeur |

**Table des matières**

**1 -Versions 3**

**2 -Introduction 4**

2.1 -Objet du document 4

2.2 -Références 4

**3 -Architecture Technique 5**

3.1 -Composants généraux 5

3.2 -Application Web 5

**4 -Architecture de Déploiement 6**

4.1 -Serveur de Base de données 6

4.2 -Serveur Debian 9 6

**5 -Architecture logicielle 7**

5.1 -Principes généraux 7

*5.1.1 -Les couches 7*

*5.1.2 -Les modules 7*

*5.1.3 -Structure des sources 7*

5.2 -Application Web 8

**6 -Points particuliers 9**

6.1 -Fichiers de configuration 9

*6.1.1 -Application web 9*

6.2.1.1 -Datasources 9

6.2 -Ressources 9

6.3 -Environnement de développement 9

6.4 -Procédure de packaging / livraison 9

**7 -Glossaire 10**

1 - Versions

|  |  |  |  |
| --- | --- | --- | --- |
| **Auteur** | **Date** | **Description** | **Version** |
| Delomez Matthieu | 10/06/2019 | Création du document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

2 - Introduction

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OCSystem.

Objectif du document permet de formaliser les étapes préliminaire du développement du système afin de rendre le développement de l'application le plus fidèle aux besoins de l'utilisateur.

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants:

* **DCF - Projet 12** : Dossier de conception fonctionnelle de l'application

3 - Architecture Technique

3.1 - Composants généraux

Une architecture doit répondre avant tout à un besoin fonctionnel donné, qui va guider les principaux choix et la structure. Les cas d’utilisation, ainsi que les principales contraintes fonctionnelles présentées dans ce document, synthétisent le besoin fonctionnel déterminant les choix d’architectures.

*Modularité*

Le système doit être modulaire afin de maîtriser sa conception initiale et sa maintenance par la suite.

*Applications de standards*

Les spécifications du système doivent être basées sur des standards techniques existants ou sur des solutions standard incluant :

· protocoles de communication (http)

· spécification des interfaces de services (Composants J2EE)) ;

· formats de représentation externe des données (XML) ;

*Orientation service*

L’orientation service fait référence à l’architecture SOA (Architecture Orienté Service), devenue une orientation de fond des SI (Système d’Information) actuels. Le futur système doit identifier une couche d’interaction de niveau service exposant ses fonctionnalités à ses propres modules (de présentation) et constituant également un point d’interopérabilité du système.

3.2 - Application Web

La pile logicielle est la suivante :

* Application **J2EE** (JDK version 1.9) /
* Serveur d'application **JOnAS 5.2.4 / Testé sur Tomcat 9.0**



- <<recherche>>: Prendra en compte tout le processus de recherche proposé par l'application, avec l'id de la bière, le bar dans lequel se trouve cette dernière, sa description etc... Requête qui sera envoyé en base et qui aura pour réponse de générer le resultat de la recherche via une Map qui indiquera tout les bars ou la bière recherché est disponible.

- <<librairie>>: Bibliothèque statique ou dynamique contenant les bières différentes ainsi que les bars, qui sera mis à jour après les services par l’administration.

-<<interface utilisateur>>: Sera en charge de tout le côté graphique de l'application, il se voudra agréable à l’oeil et ergonomique pour que l'utilisateur puisse rechercher sa demande de la manière la plus facile possible.

4 - Architecture de Déploiement



4.1 - Serveur de Base de données

PostgreSQL est un moteur de bases de données relationnelle, et un moteur adapté à des bases métier.

Caractéristiques techniques : Serveur Debian 9 + PostgreSQL 9.6 + Scripts à déployer sur PGAdmin III



4.2 - Serveur Debian 9

Debian est aujourd'hui l'une des distribution les plus reconnues, et complètes du monde Linux.

Aucun service (sauf ssh) n'est mis en place pour permettre à chacun de personnaliser son serveur à son goût.

Configuré avec OVH.

5 - Architecture logicielle

5.1 - Principes généraux

OS: Windows, Linux, Mac

Java: 1.9

Gestion de production: Spring Boot 2.1.1

Environnement Serveur : Gnome 3.8

Environnement Client : Windows / Linux / Mac

* Les sources et versions du projet sont gérées par **Git**, les dépendances et le packaging par **Spring Boot**

*5.1.1 - Les couches*

L'architecture applicative est la suivante :

* unecouche **business** : responsable de la logique métier du composant
* unecouche **model** : implémentation du modèle des objets métiers
* une couche **consumer** : dialogue avec la base de donnée et implémentation des mappers
* une couche **application**: intégration du service ainsi que les classes response necessaire au fonctionnement de l'application

Couches qui seront au sein d'un seul module Spring boot.

*5.1.2 - Les modules*

Module Spring Boot

*5.1.3 - Structure des sources*

La structuration des répertoires du projet suit la logique suivante :

* les répertoires sources sont crées de façon à respecter la philosophie Maven (à savoir : « convention plutôt que configuration »)

racine  
 ├─ *pom.xml*  
 ├─ <ModuleApplicatif>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ resources  
 └─ src  
 └─ lib

5.2 - Application Web

.Module applicatif: Correspondra à toute la partie vue de l'application, nottament les pages html, css et javascript.

6 - Points particuliers

6.1 - Fichiers de configuration

* **Logback.xml** : fichier de configuration des logs
* ***Datasources***

PostgreSql latest

* ***Fichier config.properties***

Fichier de configuration de l'application

6.2 - Ressources

Image: /src/main/recources/template/img

javascript: /src/main/recources/template

Css: /src/main/recources/template/css

6.3 - Environnement de développement

NetBeans 8.2

6.4 - Procédure de packaging / livraison

Pour faciliter l'étape de déploiement d'un projet, nous allons rassembler les différents fichiers dans une archive, mise à part la compression. Les pages html et les éléments de présentation sont packagés dans une archive WAR.

Pour déployer l'application, il vous suffit de déposer l'archive WAR dans le répertoire de notre serveur d'application.

7 - Glossaire

|  |  |
| --- | --- |
|  |  |
|  |  |