IT Consulting & Development

OC Pizza

Dossier de conception technique

Version 3

IT Consulting & Development
3 rue des Alouettes 92100 Boulogne Billancourt
0134003400
hello@itconsultingdevelopment.com
www.it-consulting-development.com
S.A.R.L. au capital de 1 000,00 €
enregistrée au RCS de 3456 SIREN 999 999 999 – Code APE : 6202A

Auteur
Alexandra Alsen Dykler
Data-analyste

IT Consulting & Development

Table des matières

1-VERSIONS

Auteur	Date	Description	Version
Alexandra Alsen Dykler	14/09/2020	Création du document	1
Alexandra Alsen Dykler	26/05/2020	Ébauche et création du document	2
Alexandra Alsen Dykler	01/06/2021	Version final	3

2-INTRODUCTION

Objet du document :

Le présent document constitue le dossier de conception technique de l'application destinée à OC Pizza.

L'objectif de ce document est de décrire la conception technique des éléments logiciels (les modules, les données, éléments de configuration, etc).

Références

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

1. Dossier des spécifications fonctionnelles
2. Dossier d'exploitation
3. Modèle physique de données

3-ARCHITECTURE TECHNIQUE

Application web :

Pour OC Pizza, nous nous sommes arrêtés sur une architecture simple à savoir une application web et une base de données. L'application web est un logiciel applicatif hébergé sur un serveur et accessible via le navigateur web préféré de l'utilisateur. L'avantage de cette solution est que l'utilisateur n'a pas besoin de l'installer sur son ordinateur.

Source

Base de données :

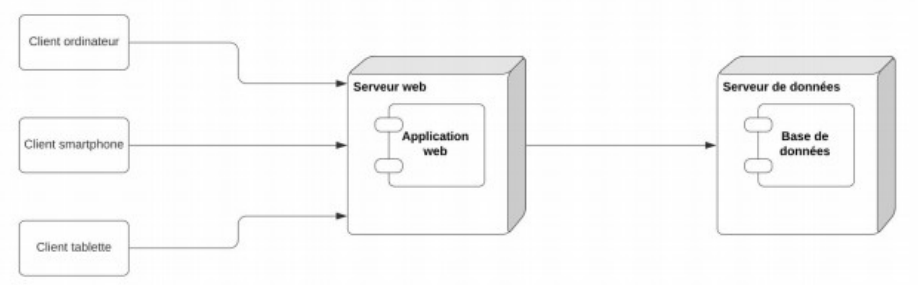
Une base de données (BDD en abrévé) est une collection d'informations organisées afin d'être facilement consultables, gérables et mises à jour. Au sein d'une database, les données sont organisées en lignes, colonnes et tableaux.

Source

Diagramme de composant :

4-Architecture de Déploiement

Diagramme de déploiement :



Serveur de bases de données :

Comme dit plus haut, la base de données est une collection d'informations organisées.

Quels sont les avantages de la base de données ?

1. Les bases de données permettent aux utilisateurs de centraliser et partagés leurs informations à tout moment
2. Les bases de données sont bien plus efficaces lorsqu'il s'agit de retrouver des informations
3. Les bases de données organisent vos informations en listes
4. Les bases de données peuvent contenir autant de données que vous le souhaitez et vous permettent de présenter ces données comme bon vous semble

Quels sont les désavantages de la base de données ?

1. Un investissement de départ relativement plus coûteux (incluant les coûts supplémentaires pour le matériel)
2. Plutôt moins efficace pour les logiciels spéciaux
3. Nécessite des employés qualifiés (administrateurs de bases de données)
4. Une vulnérabilité importante du fait de la centralisation des données

Serveur web :

Comme dit plus haut, l'application web est un logiciel applicatif hébergé sur un serveur et accessible via le navigateur web préféré de l'utilisateur.

Quels sont les avantages d'un serveur web ?

1. Le serveur web sécurise le réseau et l'ordinateur de l'utilisateur
2. La possibilité de configurer ce dernier selon ses besoins et ses préférences
3. Le recours à un serveur dédié virtuel fait gagner de la place puisque les besoins en serveur physique sont moindre

5-ARCHITECTURE LOGICIELLE

Les principes généraux :

→ Les sources et versions du projet sont gérées par **Git**

→ Les dépendances et le packaging par **Apache Maven**

→ Il y a aussi l'utilisation du framework **Springboot**

Liens utiles :

→ Accédez à git

→ Accédez à Apache Maven

→ Accédez à Springboot

Guides utilisateurs :

→ Guide utilisateur Git

→ Guide utilisateur Apache Maven

→ Guide utilisateur Springboot

Les couches métiers :

L'architecture applicative est la suivante :

1. La couche **Controller**: sert à gérer les routes
2. La couche **Model** : éléments qui modélisent les données
3. La couche **Repository** : correspond à la couche d'accès aux données
4. La couche **Service** : permet de gérer les traitements métiers

La structure des sources :

La structuration des répertoires du projet suit la logique suivante :

```
OCPizzaWeb
|-pom.xml
|-src
|    |_main
|          |_java
|          |_com.ocpizza.controller
|          |_com.ocpizza.model
|          |_com.ocpizza.repository
|          |_com.ocpizza.service
|      |_resources
|          |_applications.propoperties
|-src
|  |_test
|    |_java
|        |_com.ocpizza
```

6-POINTS PARTICULIERS

La gestion des logs :

La demande initiale d'OC Pizza étant assez simple, la gestion se fera avec ces niveaux de log :

- **Debug** : permet d'obtenir des informations simples et claires afin de déboguer une application
- **Error** : permet de désigner les événements problématiques tout en permettant l'application de continuer à fonctionner
- **Info** : permet d'afficher des messages d'information

Le fichier de configuration (application.propoperties) :

→ ce fichier sert entre autre à rentrer les informations de connexion vers la base. Vous trouverez un exemple (capture image) d'un fichier application.propoperties.

```
spring.datasource.url=jdbc:postgresql://localhost:5432/climbing
spring.datasource.username=postgres
spring.datasource.password=alex160383
spring.jpa.show-sql=true
server.port=9001
spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.security.SecurityAutoConfiguration

```

Environnement de développement :

Utilisez l'IDE Eclipse

→ Pour télécharger Eclipse : ici

→ Guide utilisateur : ici

Utilisez la dernière version de Maven

→ Maven téléchargeable : ici

→ Guide utilisateur : ici

Utilisez la version 11 de Java (JDK)

→ Java téléchargeable : ici

→ Guide utilisateur : ici

Procédure de livraison (packaging) :

Pré-requis :

- JDK11
- L'IDE Eclipse
- Maven installé sur votre machine
- Un gestionnaire de données (PgAdmin par exemple)

Actions à effectuer :

1. Télécharger le fichier export SQL de la base données et le sauvegarder dans le répertoire de votre choix
2. Importer le dans un gestionnaire de base de données en effectuant ces manipulations (ici réalisé avec PgAdmin) :

- Se connecter au serveur
- Créer une base de données vierge (clic droit, Create, Database) sauf si elle existe déjà.
- Faire un clic droit sur la base de données, Restore.
- A l'onglet General, dans le champ Filename, charger le fichier depuis le disque du serveur local ou un partage du réseau.

3. Cloner le répertoire GitHub
4. Se placer dans le répertoire du projet Maven, là où se trouve le pom.xml. Lancer la commande mvn eclipse:eclipse.
5. Importer le projet en faisant : Import > Existing Maven Projects > Browse > Select file > Finish
6. Ouvrir le fichier application.properties et saisir les données
7. Faire run as, maven build, lancer la commande clean package
8. Le fichier JAR est disponible dans le répertoire target du projet

7-Glossaire

- **Git :** Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2 (SOURCE).
- **Spring** : En informatique, Spring est un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests (SOURCE).
- **Maven** : Apache Maven (couramment appelé Maven) est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. Maven est géré par l'organisation Apache Software Foundation. L'outil était précédemment une branche de l'organisation Jakarta Project (SOURCE).
- **Eclipse** : Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java (SOURCE).
- **Diagramme de composants** : Le diagramme de composants décrit l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les composants (qui utilise quoi) (SOURCE).
- **Diagramme de déploiement** : En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype (SOURCE).