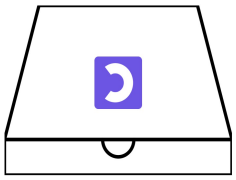


SKIT PRODUCTION

OC Pizza

Système de gestion de pizzerias

Dossier de conception technique

Version 1.0

Auteur
Sébastien Kothé
Analyste programmeur

TABLE DES MATIÈRES

1 - VERSIONS

Auteur	Date	Description	Version
Sébastien KOTHE	20/02/2021	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique du projet OC Pizza. Il est destiné à la maitrise d'ouvrage (MOA) et à la maitrise d'oeuvre (MOE).

L'objectif de ce document est de permettre aux concepteurs du site web de réaliser leur mission dans les meilleures conditions possibles.

Les éléments du dossier ont été élaborés à partir du document de spécifications fonctionnelles réalisé par SKIT Production.

2.2 - Références

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

1. **DCF – 1.0**: Dossier de conception fonctionnelle de l'application
2. **DE – 1.0**: Dossier d'exploitation
3. **PVL – 1.0**: Procès verbal de livraison

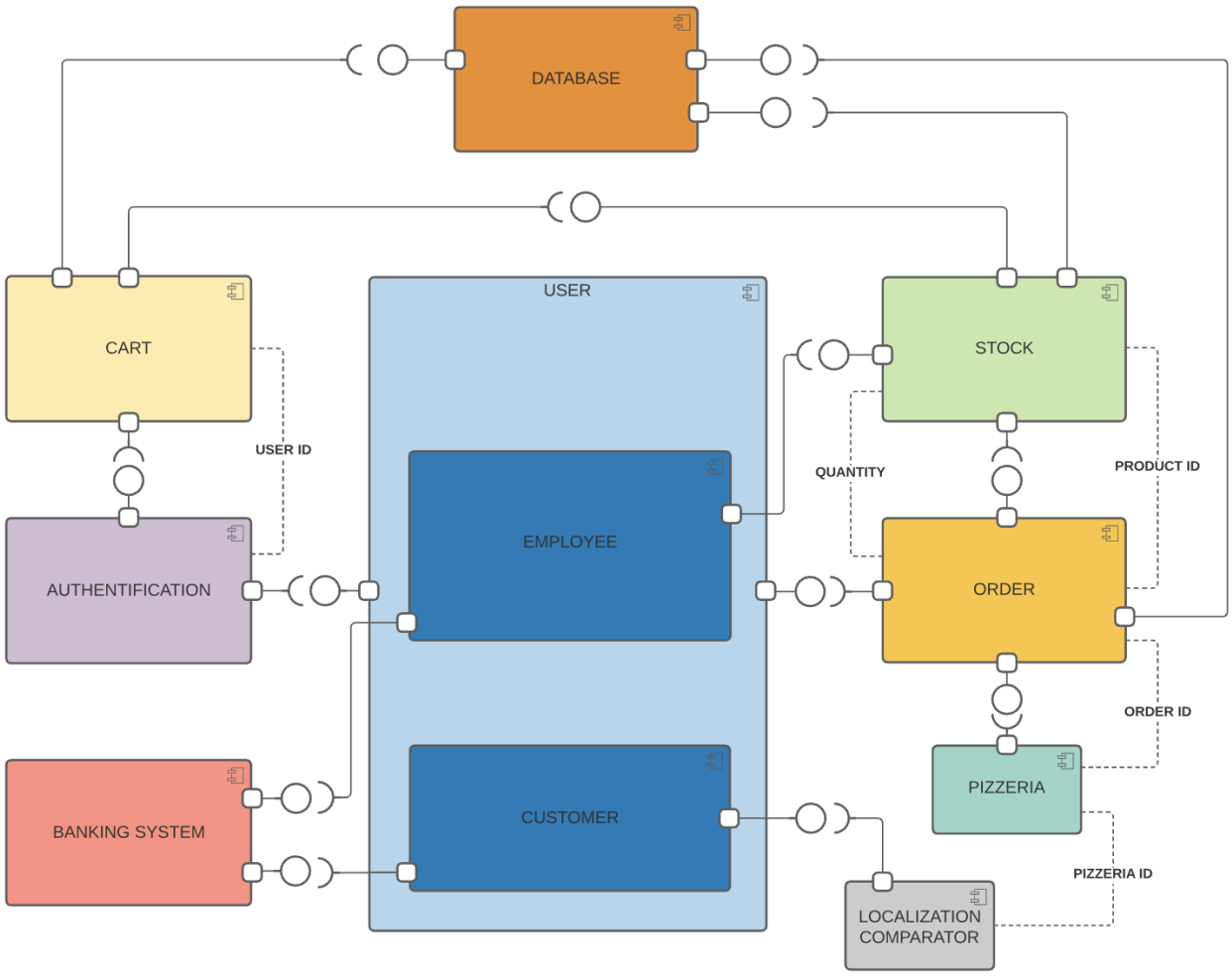
3 - Architecture Technique

Comme nous l'avons indiqué dans le dossier des spécifications techniques, le site web sera crée à l'aide du langage Swift et du framework Vapor. Au sein du code nous utiliserons Leaf qui est un langage de templates permettant de transformer du code Swift en code HTML ou JSON, exploité par le navigateur web ou le service web. Leaf génère ce code directement depuis le serveur. Nous utiliserons également Fluent qui est un ORM (object-relational mapping) pour Vapor qui permet de transformer les requêtes et les données des gestionnaires de base de données (MySQL, PostgreSQL) en objets exploitables directement dans le code.

3.1 - Les composants

Le diagramme de composants permet d'illustrer la relation entre les différents composants du sytème. Il montre également les dépendances entre les composants: un composant avec une interface requise a besoin d'un autre composant du système pour la lui fournir. Exemple: STOCK, ORDER et CART nécessite les interfaces fournies par le composant DATABASE, à l'image de la dépendance entre le composant USER et BANKING SYSTEM. Tous les composants sont dits « internes » à l'exception de BANKING SYSTEM et DATABASE qui sont dits « externes ».

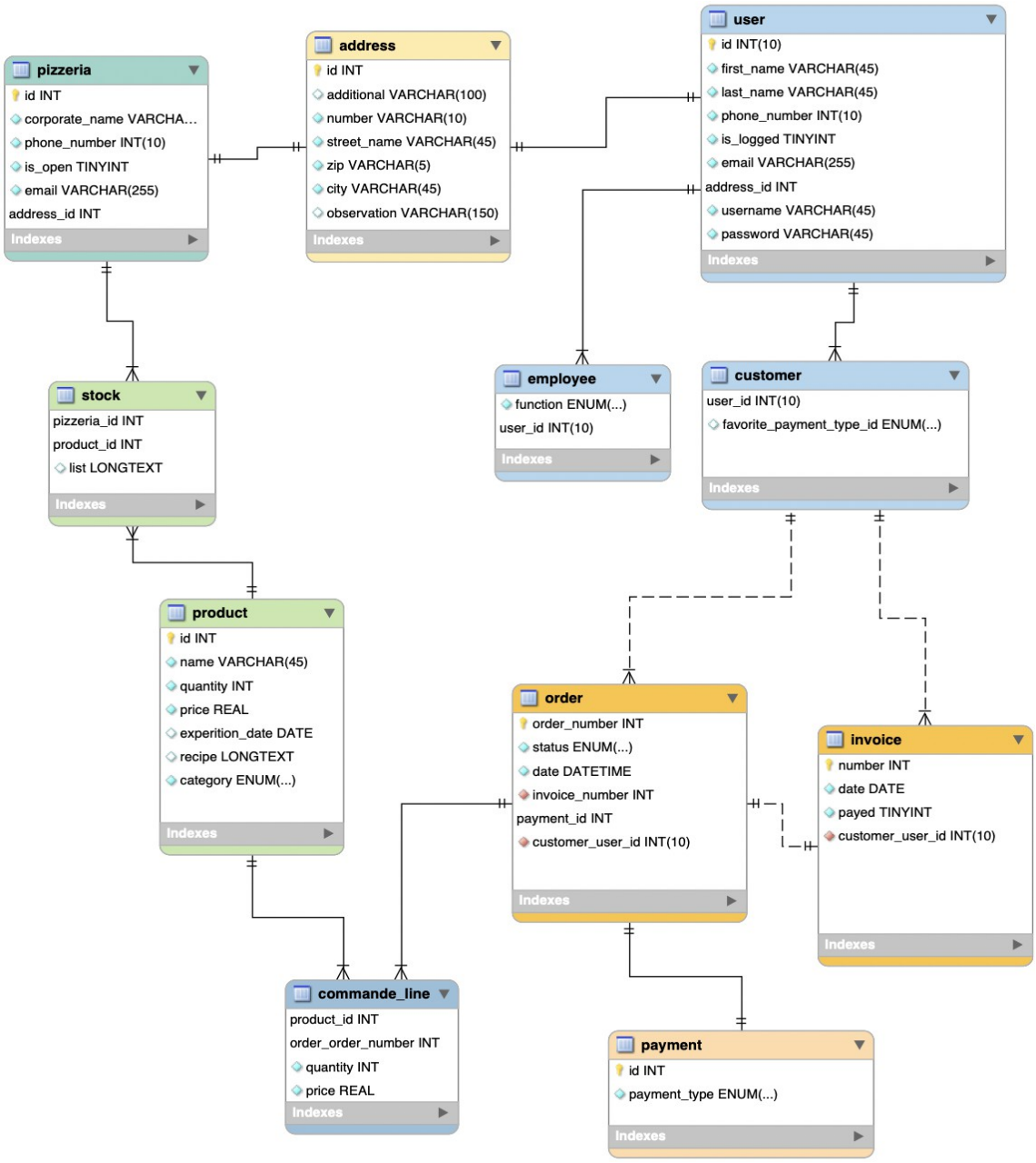
Notez que les traits pointillés indiquent les informations qui transitent d'un composant à l'autre.



3.2 - Modèle physique de données

Elaboré à partir du diagramme de classe, le modèle physique de données est une représentation graphique d'une base de données. Il contient des attributs typés, introduit la notion de clé primaire/étrangère et les tables sont liées entre elles par des relation identifiantes ou non-identifiantes. Les tables possèdent des clés primaires, souvent auto-incrémentées, qui portent le plus souvent le nom « id », ou bien c'est un attribut de la table qui peut prendre ce rôle comme pour « number » dans la table INVOICE. La clé primaire des tables d'associations comme STOCK, ont une clé primaire composée: STOCK possède une clé primaire composée des clés étrangères de PIZZERIA et PRODUCT. Notez que les losanges vides qui précèdent les attributs, indiquent que ces derniers sont optionnels.

C'est à partir du MPD que l'on peut générer le script SQL pour créer la base de données. Cette opération est effectuée via MySQLWorkbench en faisant un forward engineer.



4 - Architecture de Déploiement

Le site sera déployé sur le cloud avec Heroku qui donne une plateforme de production clés en main.

La location du nom de domaine sera faite auprès du bureau d'enregistrement gandi.net. Après enregistrement, il faudra faire quelques réglages pour que le domaine ocpizza.com pointe vers Heroku:

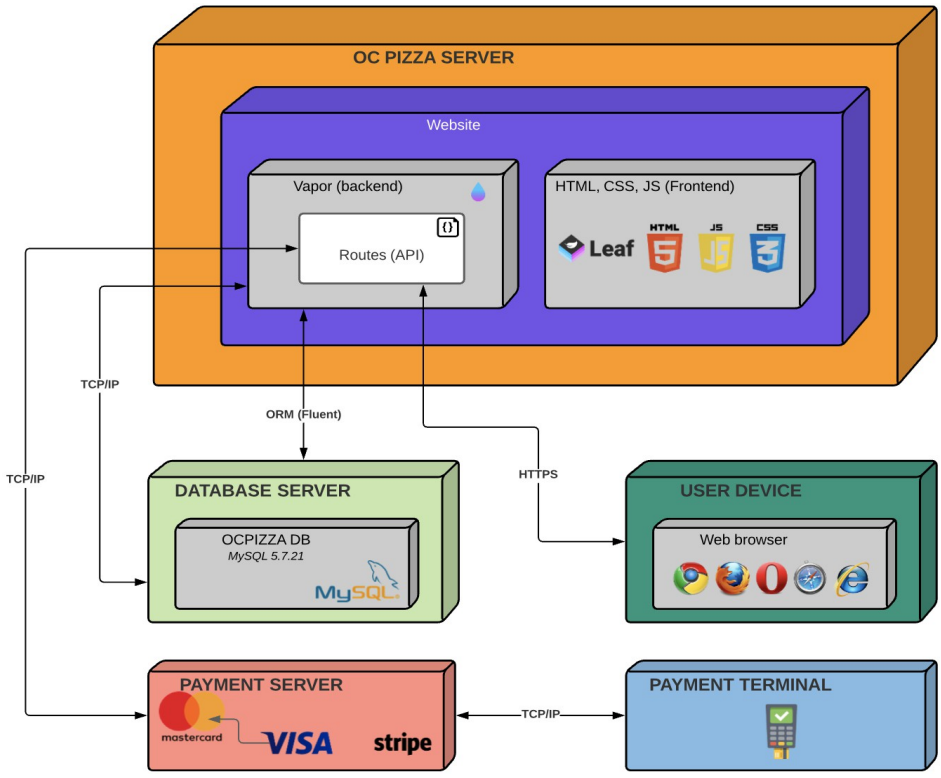
1) Ajouter le nom de domaine depuis le site Heroku et copier la « DNS Target » (*deep-mangosteen-bfueaxscjoq0opqm8y7qhdbd.herokudns.com*)
2) Coller cette dernière dans la partie DNS Management du site gandi.net

La procédure est détaillée précisément ici: Add a custom domain. Vous pouvez vérifier si la configuration est effective avec cette commande:

host www.ocpizza.com

---> www.ocpizza.com is an alias for *deep-mangosteen-bfueaxscjoq0opqm8y7qhdbd.herokudns.com*

Le diagramme de déploiement ci-dessous permet de visualiser rapidement les principaux éléments constituants du système et la façon dont ils communiquent entre eux. Le site est constitué de la partie Frontend (HTML, CSS, JavaScript et Leaf) et de la partie Backend qui utilise le framework Vapor. Il est important de préciser que le système est dépendant de Stripe ainsi que de la base de données MySQL.



4.1 - Serveur de Base de données

Le choix du serveur s'est porté sur une base de données qui communique avec le SGBD (Système de Gestion de Base de Données) MySQL. Ce dernier contient les principaux instruments permettant de gérer la base de données. Il est important de préciser que MySQL est compatible avec le Cloud, ceci étant parfaitement adapté à notre projet (Heroku). Comme dit précédemment, le schéma de la base dé données a été réalisé avec MySQLWorkbench.

5 - ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

Les sources et versions du projet sont gérées par **Git** et hébergées sur **Github**.

5.1.1 - Les couches

L'architecture applicative est la suivante:

- une couche **Model**: responsable de la logique métier du composant
- une couche **Vue**: responsable de l'interface utilisateur
- une couche **Controller**: joue le rôle d'intermédiaire entre le **Model** et la **Vue**

5.1.2 - Structure des sources

Les répertoires sources sont crées de façon à respecter l'architecture MVC

```
racine
├─ Model
│  ├─ Food
│  │  ├ Product
│  │  ├ Ingredient
│  │  └ Category
│  ├─ Restaurant
│  │  ├ Pizzeria
│  │  └ Employee
│  ├─ Client
│  │  ├ Customer
│  │  └ Invoice
│  ├─ AssociationClasses
│  │  ├ CommandLine
│  │  └ Stock
│  ├─ SharedClasses
│  │  ├ Order
│  │  ├ Payment
│  └ └ └ Address
│
├─ View
│  ├─ Menu
│  ├─ Authentication
│  ├─ Profile
│  └─ Order
│
├─ Controller
│  ├─ Menu
│  ├─ Authentication
│  ├─ Profile
└ └ └ Order
```

6 - POINTS PARTICULIERS

6.1 - Gestion des logs

Les logs permettent de garder une trace de ce qui se passe sur l'application. Souvent ils révèlent une source très précieuse d'informations. Les éventuelles erreurs et dysfonctionnements du serveur Gandi sont gérés par la plateforme dédiée.

6.2 - Environnement de développement

L'environnement de développement du site web sera l'IDE (Integrated Development Environment) Xcode sous le système d'exploitation macOS Big Sur (version 11.2.1) en important le framework Vapor (Version 4.9.0).