

OC Pizza

**OC Pizza**

# **Order management application**

Technical Design file

**Spencer Forrest**

## TABLE OF CONTENTS

<b>1 - Versions .....</b>	<b>3</b>
<b>2 - Introduction .....</b>	<b>4</b>
<b>2.1 - Document purpose .....</b>	<b>4</b>
<b>3 - Technical Architecture .....</b>	<b>5</b>
<b>3.1 - General components .....</b>	<b>5</b>
3.1.1 - Description .....	5
3.1.2 - Deployment diagram.....	6
<b>4 - Data Architecture .....</b>	<b>7</b>
<b>4.1 - Functional Domain.....</b>	<b>7</b>
4.1.1 - Class diagram.....	7
4.1.2 - Explanation.....	8
<b>Annex 1.....</b>	<b>9</b>

# 1 - VERSIONS

Author	Date	Description	Version
Spencer Forrest	11/06/2019	Document creation	1.0

## 2 - INTRODUCTION

### 2.1 - Document purpose

The present document makes up the technical conception folder which represents the technical solution used to implement the application. It will present the functional domain, the physical organization and the external web services which constitute the system.

## 3 - TECHNICAL ARCHITECTURE

### 3.1 - General components

#### 3.1.1 - Description

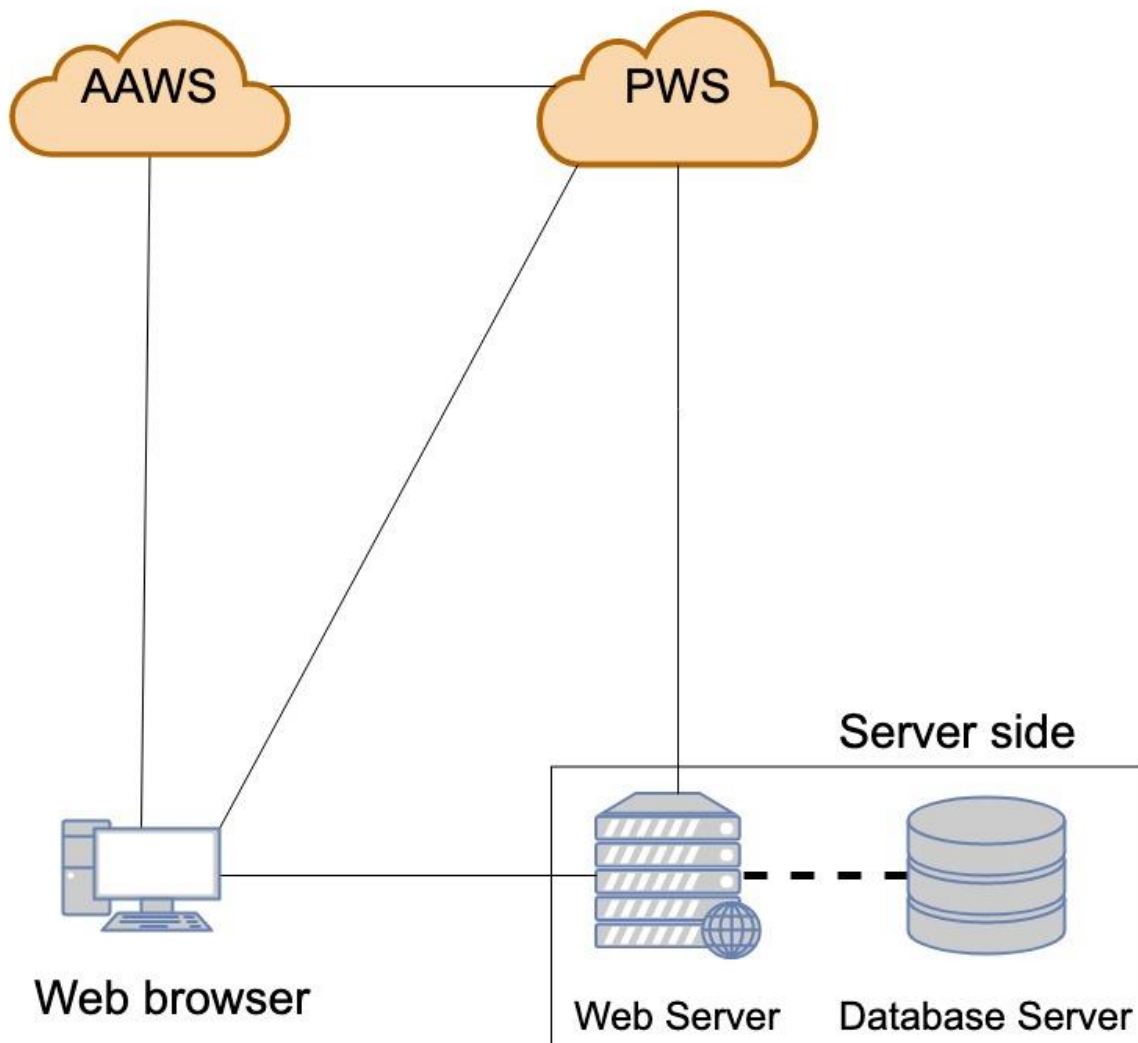
There will be three web services (see deployment diagram on next page):

- Address autocomplete (“**Place Autocomplete service**” from google)
- Payment web service (“**Stripe**”)

All web services are REST APIs using JSON in order to communicate with mobile applications since their frameworks facilitate the use of JSON.

This architecture will use **MySQL** for the database and **Ruby on Rails** to implement the web application. The application will be deployed using the managed hosting service **Heroku**.

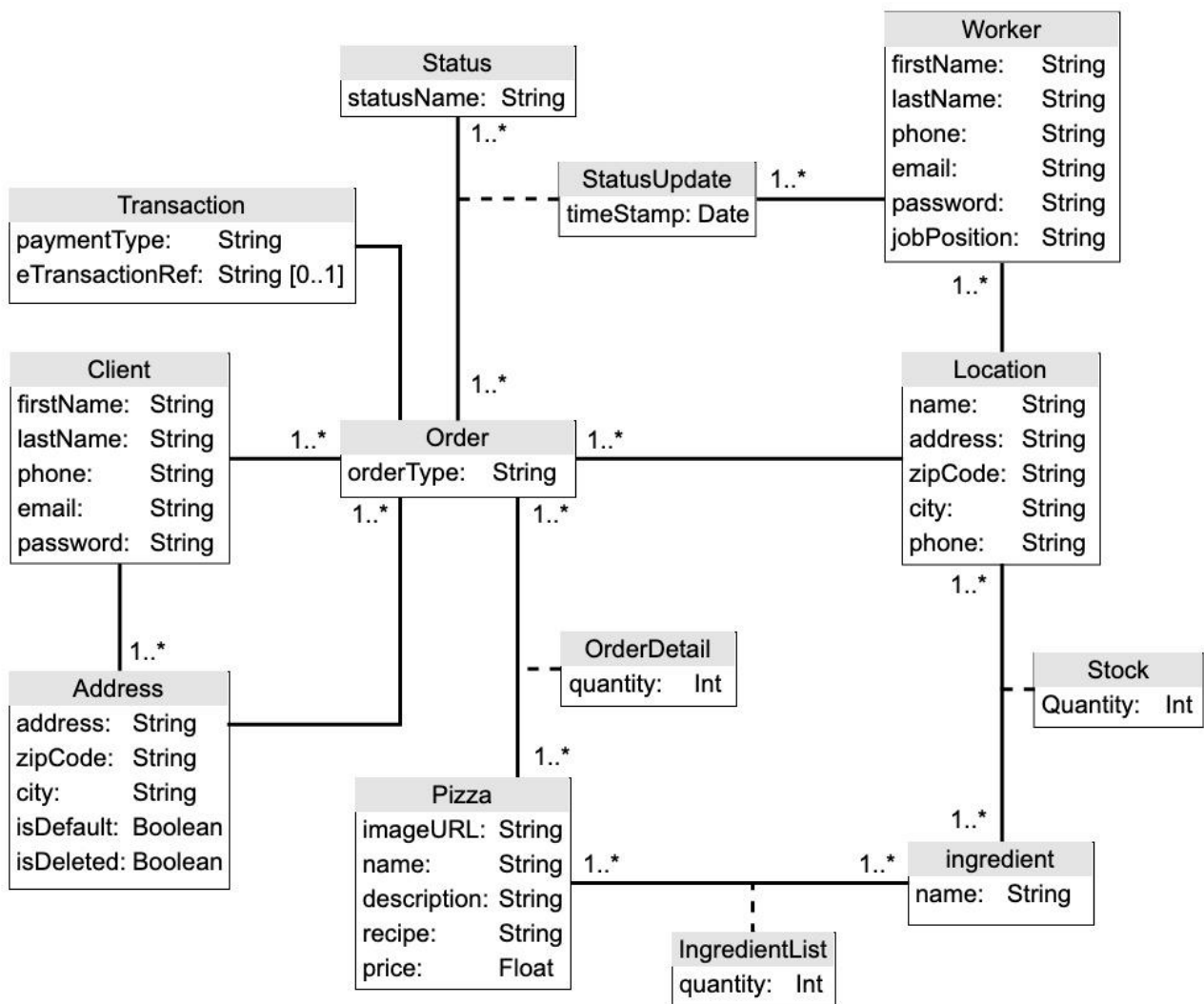
### 3.1.2 - Deployment diagram



## 4 - DATA ARCHITECTURE

## 4.1 - Functional Domain

### 4.1.1 - Class diagram



#### **4.1.2 - Explanation**

- The “Pizza” class represents a type of pizza (e.g hawaiian, pepperoni).
- The “Ingredient” class represents a type of ingredient (e.g tomato, olive).
- The “eTransactionRef” attribute in the “Transaction” class is a reference to an online transaction. It can be null since a transaction can be done online or upon delivery.
- The “StatusUpdate” class keeps track of which workers changed the status of an order and when they did it.
- The “Transaction” class captures how a user paid its order (e.g. online, cash). There can only be one transaction by order and one order by transaction.

The **Physical Data Model** (Annex 1), created from the class diagram, is the actual design blueprint of the relational database.



## ANNEX 1

