Docent Jeroen de Kort, Semester 4, 2020/2021

A close up of a logo

Description automatically generated

MARIO PIZZERIA CASUS ANALYSIS

By Lex van Dalen and Slawomir Twardowski

Table of Contents

[Introduction 2](#_Toc50031114)

[Domino’s website analysis 3](#_Toc50031115)

[Mario Pizzerias data types 8](#_Toc50031116)

[Extra ingredienten.csv 9](#_Toc50031117)

[MarioOrderData01\_10000.csv - MarioOrderData04\_10000.csv 9](#_Toc50031118)

[Overige producten.xlsx 9](#_Toc50031119)

[pizza\_ingredienten.xlsx 10](#_Toc50031120)

[pizzabodems.xlsx 10](#_Toc50031121)

[Winkels Mario.txt 10](#_Toc50031122)

[Postcode tabel.mdb 11](#_Toc50031123)

# Introduction

This document concerns the analysis of the Super Mario Pizzeria casus. At first, we’re going to analyze the Domino’s website with its functionality concerning the use of (dynamic) data fetched from databases. Screenshots will be used as a means of help in visualizing the layout and actions taking place on the site.

In the next step we will analyze the data files available as part of the current Super Mario Pizzeria operation. Those files are in several different database formats and the data contained will have to be cleaned up and transferred to a new all-encompassing database of our own design.

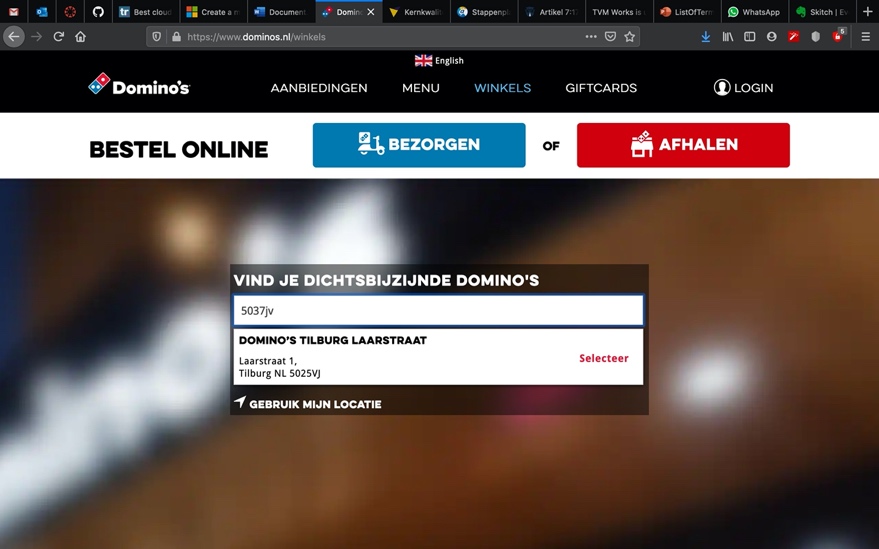
# 

# Domino’s website analysis

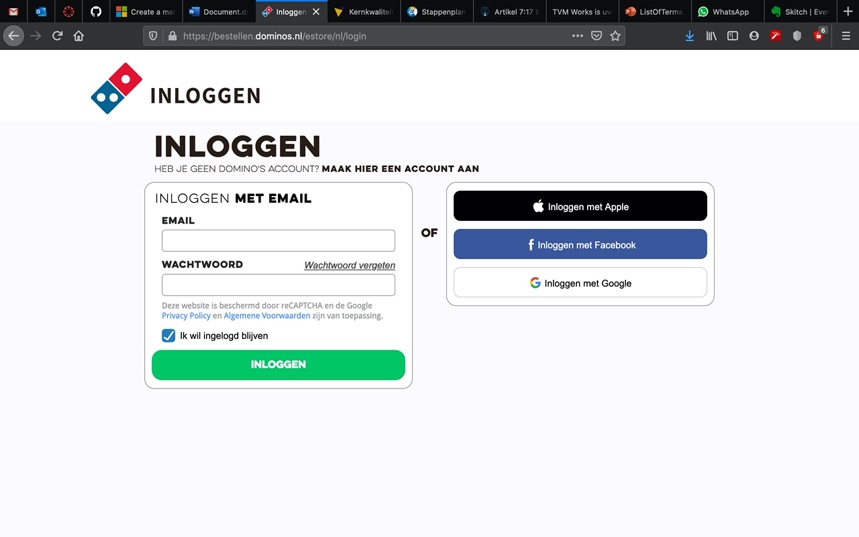
Website in question can be found under this link:

<https://www.dominos.nl/>

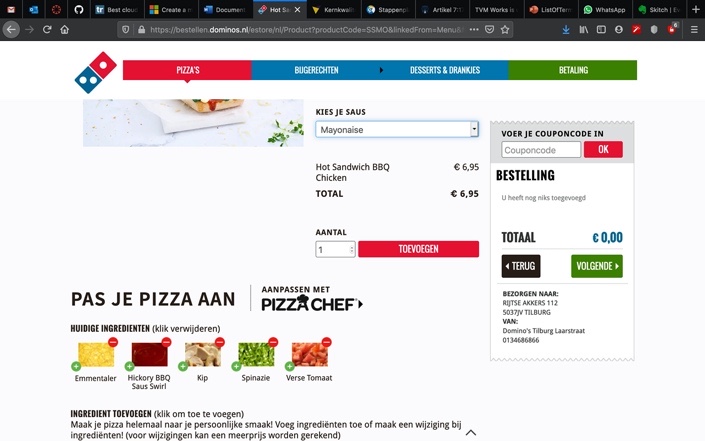
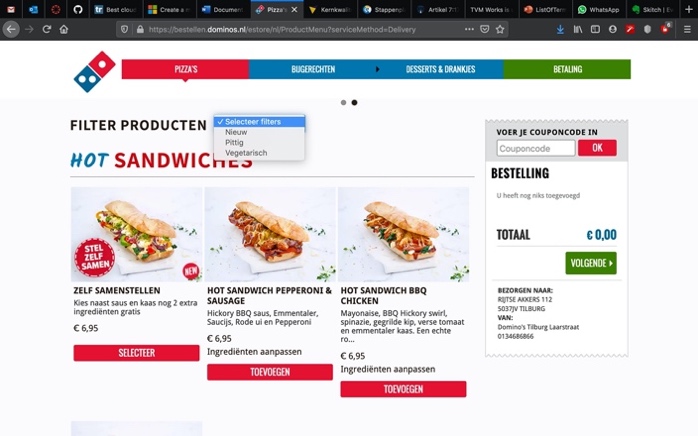
Typing in postcode automatically suggest a list of nearby pizza shops:



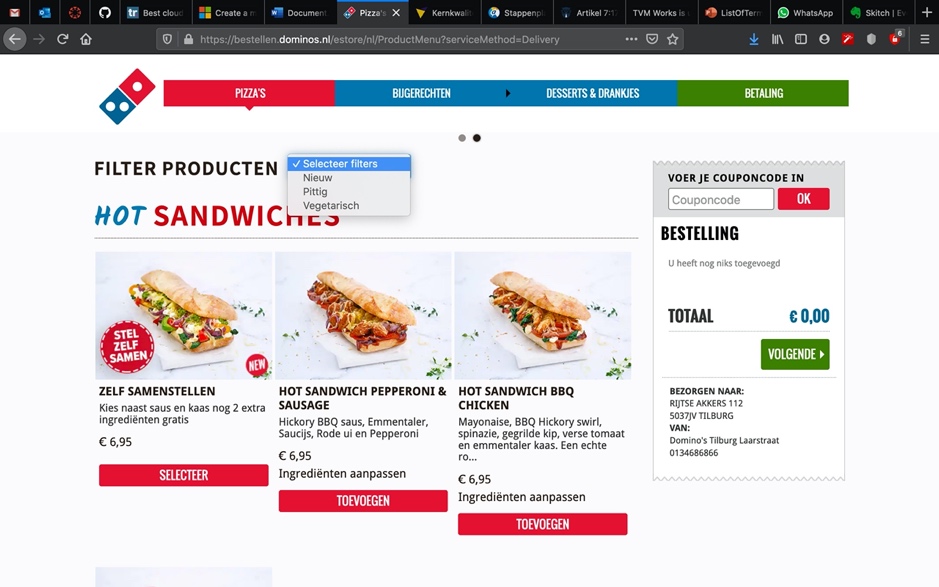
Login possibility to trace orders and recall address. Orders are being stored on the account for further use, like ordering the same meals etc.:



Menu display + option to Add product to the order. Order details like price, qty, products to buy etc. are dynamically updated in a sidebar:

Products can be filtered by selecting a specific filtering option:



The site keeps recently used addresses in cache for convenience even without the use of an account:

A screenshot of a cell phone

Description automatically generated

Selecting specific ingredients is realized by using drop down lists and the fee is being calculated dynamically as the user selects given ingredients:

A screenshot of a computer

Description automatically generated

If selected by the user pizza shop is closed at the moment it is still possible to place an order on a specified date and time:

A screenshot of a computer

Description automatically generated

Checkout requires consent to store the user data and lets the user use the option to leave a notice to the delivery person:

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

At checkout payment method can be selected. It is possible to tip the delivery person at this point and the specified tip is added to the total cost:

A screenshot of a computer screen

Description automatically generated

# Mario Pizzerias data types

There are different data file types used in the Mario Pizzerias current system. The files use different formats and are organized in different ways. This presents a challenge in aligning the databases with one another. There’s also a problem with typing errors present in the data.

Below is a list of provided files with a short description of the contents and possible obstacles.

## **Extra ingredienten.csv**

Contains a list of ingredients (like: ananas, bacon, salami etc.) in one row and price in second row. While the cost of ingredients seems to be correct the currency symbol used seems to be a corrupted one. File consists 29 fields.

## **MarioOrderData01\_10000.csv - MarioOrderData04\_10000.csv**

Those 4 files contain a list of orders placed by clients. A complete orders history is stored here. Fields used:

*Winkelnaam, Klantnaam, TelefoonNr, Email, Adres, Woonplaats, Besteldatum, AfleverType, AfleverDatum, AfleverMoment, Product, PizzaBodem, PizzaSaus, Prijs, Bezorgkosten, Aantal, Extra Ingredi√´nten, Prijs Extra Ingredi√´nten, Regelprijs, Totaalprijs, Gebruikte Coupon, Coupon Korting, Te Betalen.*

Note that some names seem to be corrupted. Each order consists of as many rows as there are products/ingredients in the order and each order is separated by an empty row.

## **Overige producten.xlsx**

Contains a list of products other than pizzas (ie. desserts). Field names:

*categorie, subcategorie, productnaam, productomschrijving, prijs, spicy, vegetarisch*

Not all products have a description (empty *productomschrijving* field). Last 2 columns are binary values. Row 23 seems to be overlapping two different categories of products (corrupted).

## **pizza\_ingredienten.xlsx**

This file contains all the pizza ingredients found on any particular pizza (similarly to a recipe). One type of pizza takes as many rows as there are ingredients on it. Fields:

*categorie, subcategorie, productnaam, productomschrijving, prijs, bezorgtoeslag, spicy, vegetarisch, beschikbaar, aantalkeer\_ingredient, ingredientnaam, pizzasaus\_standaard*

## **pizzabodems.xlsx**

A list of available pizza bottoms. Fields:

*naam, diameter, omschrijving, toeslag, beschikbaar*

## **Winkels Mario.txt**

This file contains a list of all the shops. Shops are separated by an empty line. The data seems to be organized in the following manner (top to bottom for a shop):

* *Place* (this can be a city or district, it’s inconsistent)
* *Street*
* *House number*
* *City*
* *Country code*
* *Post code*
* *Phone number*

As this document has been typed in by an employee there are some typing errors etc. which would need cleaning up.

## **Postcode tabel.mdb**

This database consists of all the postcodes and corresponding address breakdown in POSTCODES table; and municipality data in GEMEENTEN table. On this basis an address can be deducted from the postcode alone. It has the following fields:

*POSTCODES table:*

*A13\_POSTCODE, A13\_REEKSIND, A13\_BREEKPUNT\_VAN, A13\_WOONPLAATS, A13\_STRAATNAAM, A13\_GEMEENTECODE*

*GEMEENTEN table:*

*N42\_GEM\_KODE, N42\_GEM\_NAAM*