

# Documentation technique

**Prénom :** Valentin

**Nom :** Burillier

**Nom du projet :** Le-Quai-Antique

**Lien Github :** <https://github.com/ValentinBurillier/Le-Quai-Antique.git>

**Lien de l'application :** <https://obscure-waters-65708.herokuapp.com/>

**Lien gestion du travail :** <https://trello.com/b/poB6Gz1J/le-quai-antique-j-41>

## 1. Description des tâches et dans quelles conditions

Dans un premier temps je me suis mis à la place du client en voulant comprendre clairement ses objectifs et ses intentions. Le but était de réaliser un site reflétant sa personnalité et sa carrière mêlant le professionnel et la simplicité mais aussi d'être visible car il s'agissait de l'ouverture de son restaurant. Il fallait cependant réaliser ce site en 1 mois maximum et avec quelques fonctionnalités principales comme un système de connexion, une réservation en ligne et l'indication des moyens de contact du restaurant. Avant de commencer, j'ai d'abord écrits des « users story » (voir dossier annexe) pour me mettre à la place à la fois des visiteurs, mais aussi de l'administrateur qui devra gérer les données.

Une fois effectué, j'ai réalisé des recherches sur des sites concurrents (dossier annexe) pour comprendre la tendance actuelle, les besoins des restaurateurs demandant un site vitrine et appréhender le design type du site. Nous avons ainsi convenu d'utiliser un style sobre et classique reflétant son village avec des couleurs dont le marron, le blanc et le noir et une police classique (voir maquette en annexe).

J'ai réalisé ensuite une maquette / wireframe de l'application qu'ils ont d'abord appréciée. Par la suite cependant, les conceptions ayant évoluées, plusieurs pages ont été modifiées / voire supprimée pour répondre à la satisfaction client. Pour avoir des avis extérieurs et en accord avec mon client, j'ai présenté également l'application à mes proches ce qui a conduit à l'amélioration de certains effets stylistiques comme la modification de la police d'origine ou encore le retrait de l'opacité en fond.

Une fois la maquette terminée, j'ai créé plusieurs diagrammes fournis en annexe dont le diagramme de cas d'utilisation, le diagramme de paquetage et le diagramme de séquence pour avoir une structure claire des cas d'utilisations, des acteurs et des actions réalisées sur le site ; mais aussi une structure concernant les tables de ma base de données et les liens entre elles.

Au cours de la réalisation de cette application, plusieurs difficultés se sont révélées dont le temps principalement ce qui a conduit au report de fonctionnalités secondaires non indispensables à l'instant t. Une difficulté supplémentaire a été la mise en superposition d'un fond blanc sur une image de fond dans la section du menu contenant les entrées, plats, desserts et vin à cause du responsive et la superposition des deux images avec de plus le texte au-dessus qui devait être centré. Le déploiement via Heroku a été également un problème dû à une commande de migrations qui échouait en indiquant que la table a déjà été créée.

Après énormément de recherches, la seule solution que j'ai pu trouver à temps a été d'exporter chaque table de ma base de données et réécrire à nouveau les commandes SQL manuellement directement.

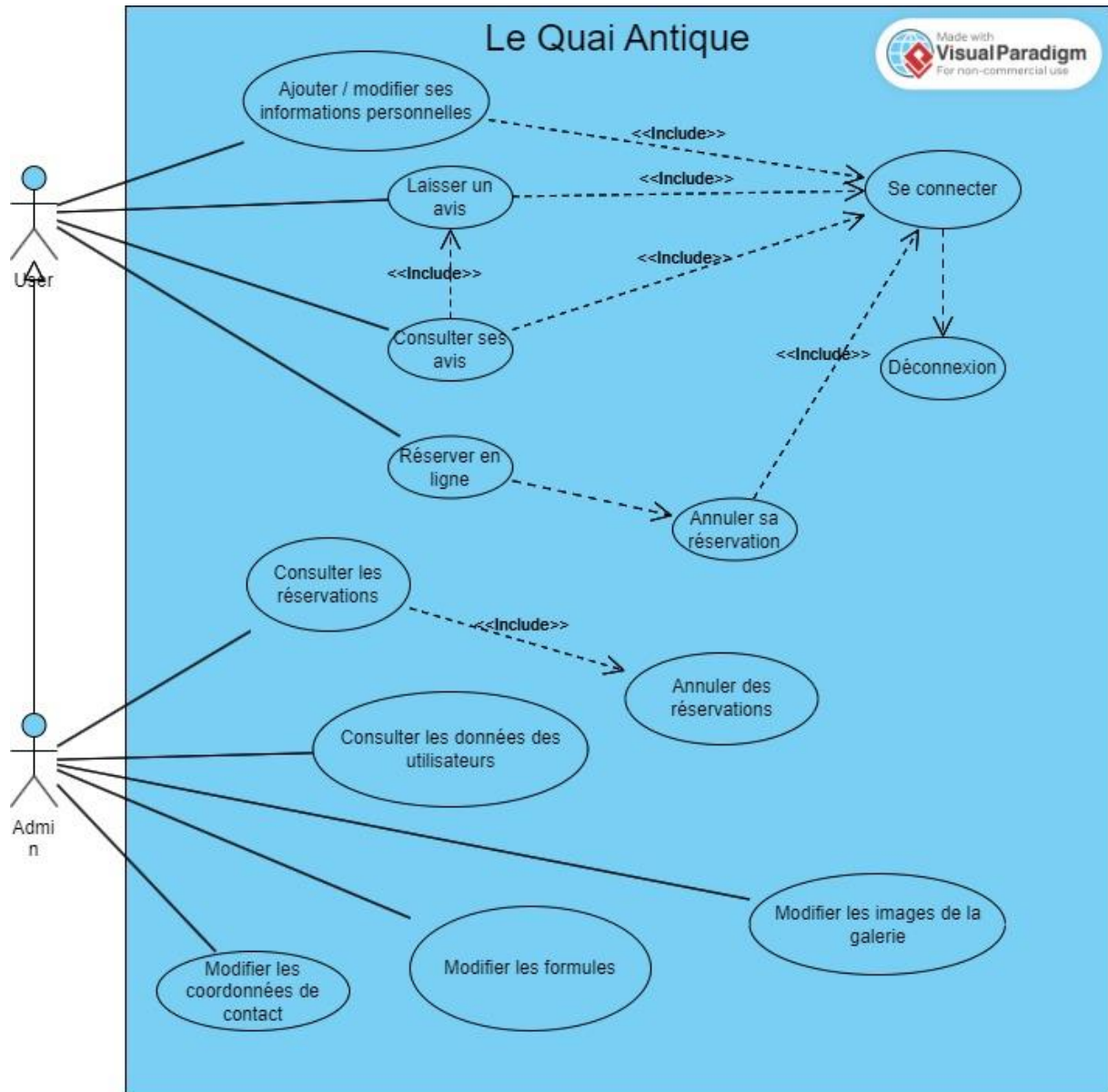
Cette application a été réalisée entièrement en télétravail. Concernant le responsive, il a été convenu de commencer sur une base de 320 x 568 px qui correspond à un petit smartphone jusqu'à un ordinateur de bureau environ 1800 x 874px.

## 2. Moyens utilisés

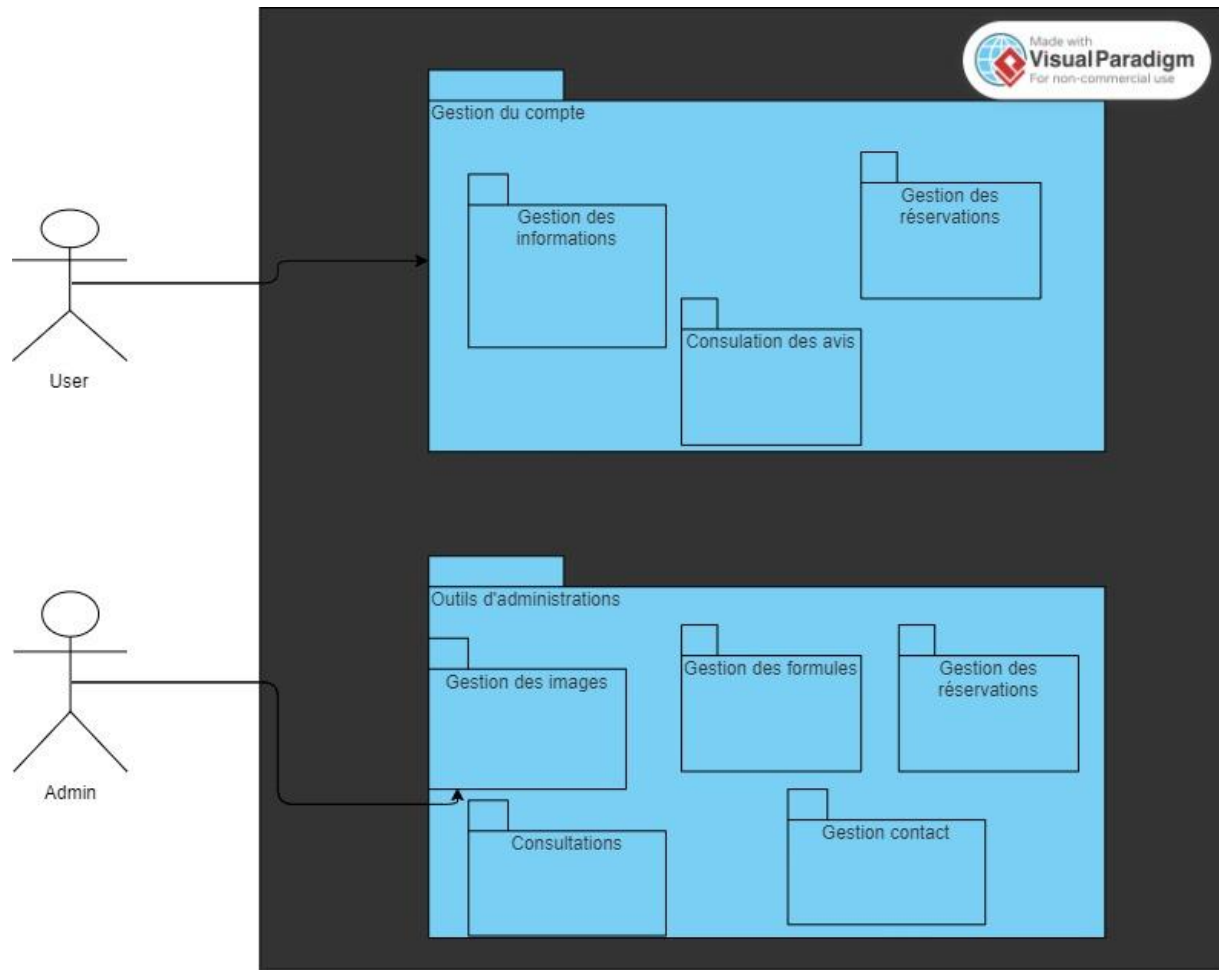
Pour réaliser cette application, je n'avais aucune contrainte de la part du client. Comme cette application comportait des données, j'ai choisi le framework Symfony basé sur le langage de programmation PHP ce qui fut très utile pour la simplicité de la création des entités. Concernant le responsive, j'ai préféré opter pour du CSS afin de personnaliser au maximum le site même si cela devait prendre un peu plus de temps afin de le rendre « unique ». Pour le style, l'utilisation du préprocesseur Sass a été très utile pour compiler, découper le code en composants réutilisables et personnalisables ; mais aussi voir apparaître les erreurs de syntaxes et debugger plus rapidement. L'écriture du code a été réalisée via l'IDE Visual Studio Code et j'ai utilisé github pour la gestion et l'évolutivité du code. Concernant la documentation, j'ai utilisé principalement la documentation en ligne pour les fonctions PHP, le déploiement avec Heroku et quelques effets CSS via Mozilla.

## Annexe

Diagramme de cas d'utilisation :

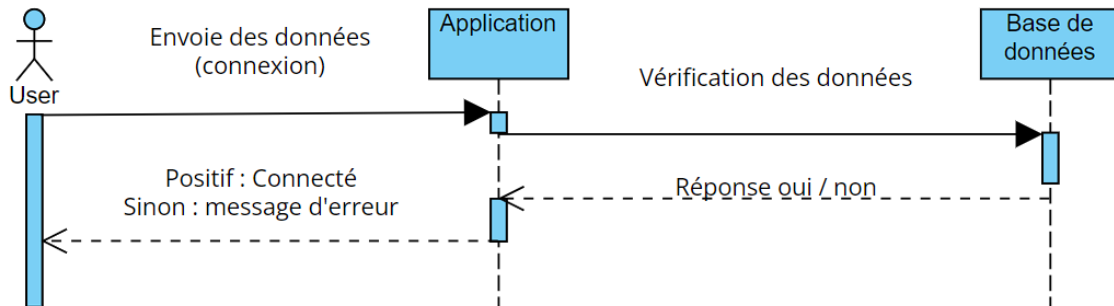


## Diagramme de paquetage

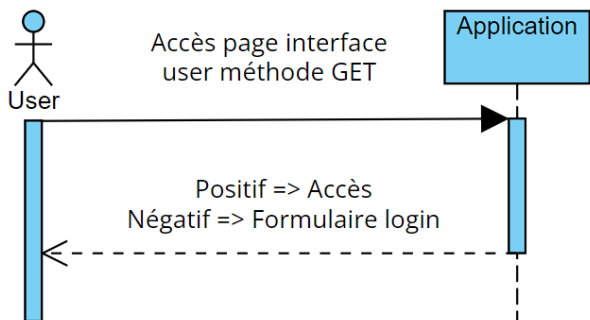


## Diagrammes de séquences

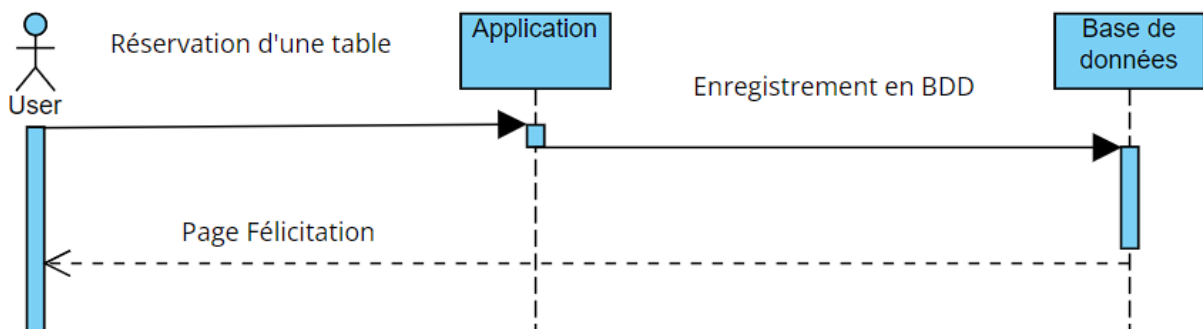
### Formulaire de connexion : login



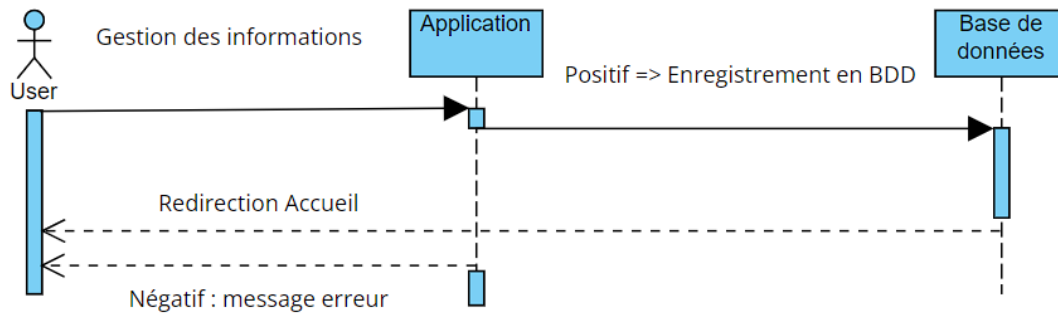
### Accès page interface « user » via méthode GET par exemple dû à l'ajout de la page en favoris



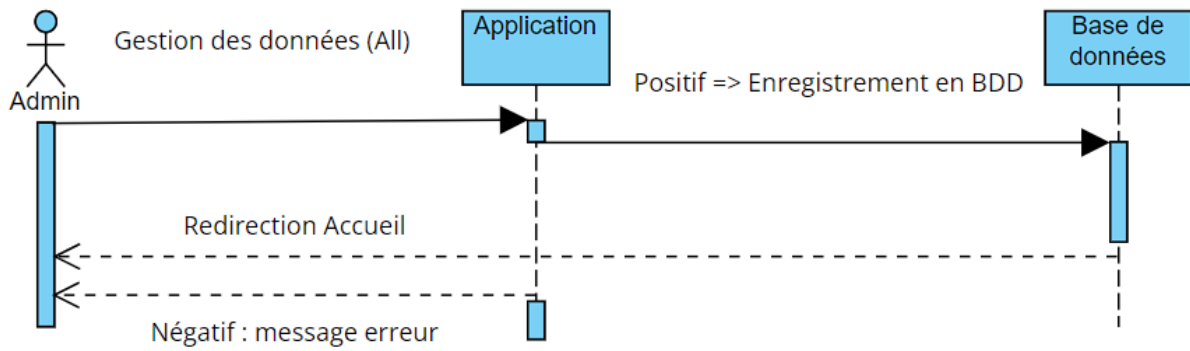
### Réservation d'une table



## Modifications des informations personnelles



## Gestion des informations par l'administrateur



# Diagramme de classes

## Class Diagrams

