

Studi ECF 2023 – 2024

DOCUMENTATION TECHNIQUE



GARAGE V.PARROT

"Réparations fiables, voitures d'occasion exceptionnelles."

DOCUMENTATION TECHNIQUE

Table des matières

SPECIFICATION TECHNIQUES	page 3
DIAGRAMME DE CAS D'UTILISATION.....	page 4
DIAGRAMMES DE SEQUENCES.....	page 5
US1 – Se connecter	
US4 – Filtrer les véhicules d'occasion	
US5 – Permettre de contacter l'atelier	
US6 – Recueillir les témoignages des clients	
DIAGRAMME DE CLASSE.....	page 6
MESURE DE SECURITE	page 7
L'authentification	
Protection contre les attaques de forces brutes	
Les autorisations	
Protection contre les injections SQL	
Les formulaires	
Le HTTPS (HyperText Transfer Protocol Secure)	
DEPLOIEMENT SUR HEROKU.....	page 9
Installation et configuration de la base de données JawsDB MySQL	
Configuration de la boîte mail MAILTRAP	

SPECIFICATION TECHNIQUES

FRONT

- HTML 5
- CSS 3
- TWIG
- Javascript
- JQuery (3.7.0)

SERVEUR LOCAL

XAMPP (8.2.4)

- Maria DB (10.4.28)
- Apache/2.4.56
- PHP (8.2.4)

BACK

- Composer (2.5.8)
- Symfony (6.3)
- Bundles :
 - Doctrine
 - VichUploader
 - Knp-paginator (6.2)
 - Form
 - Make-bundle (dev)
 - Security bundle
 - Password-hasher
 - Validator
 - Annotations
 - Rate-limiter
 - Profiler (dev)
 - Extension intl (heroku)

PRODUCTION

HEROKU (8.4.2)

- MariaDB
- Apache
- PHP
- JawsDB MySQL
- Mailtrap

ENVIRONNEMENT DE TRAVAIL

- Windows 10
- Visual Studio Code
- Figma (Wireframes & Mockup)
- Draw.io Integration version 1.6.6 (UML)
- Github
- Trello

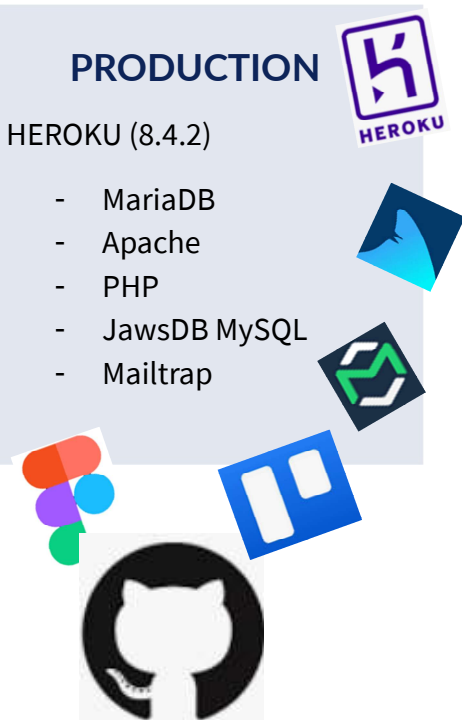
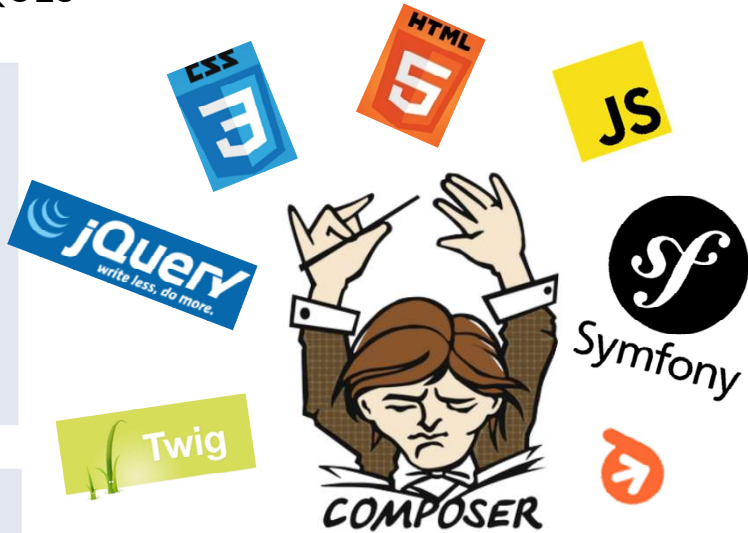


Diagramme de cas d'utilisation - Garage V. PARROT

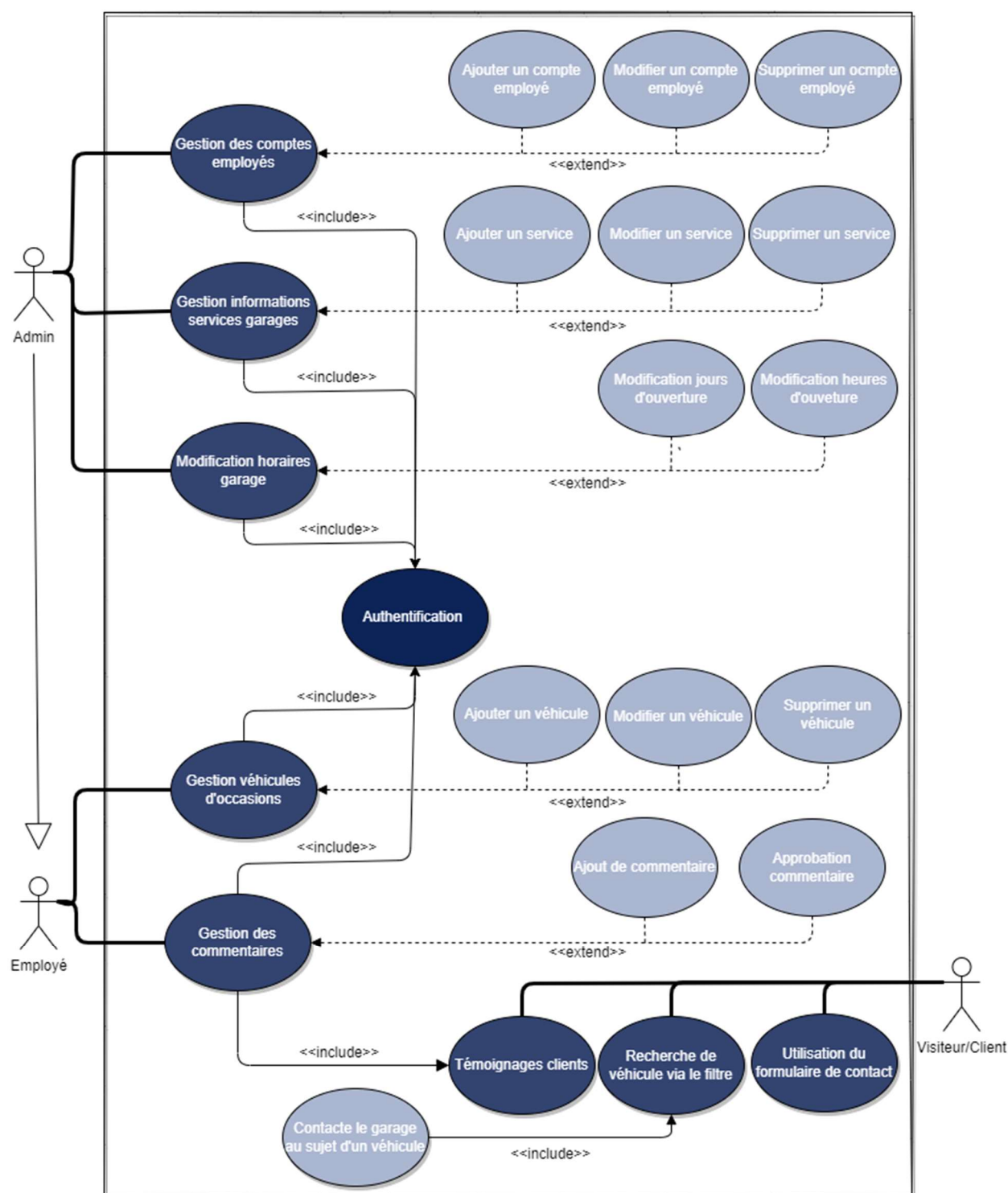
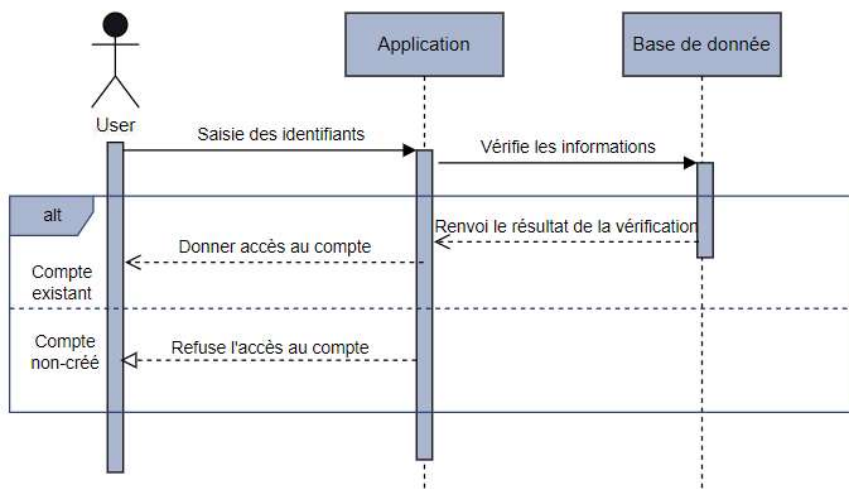
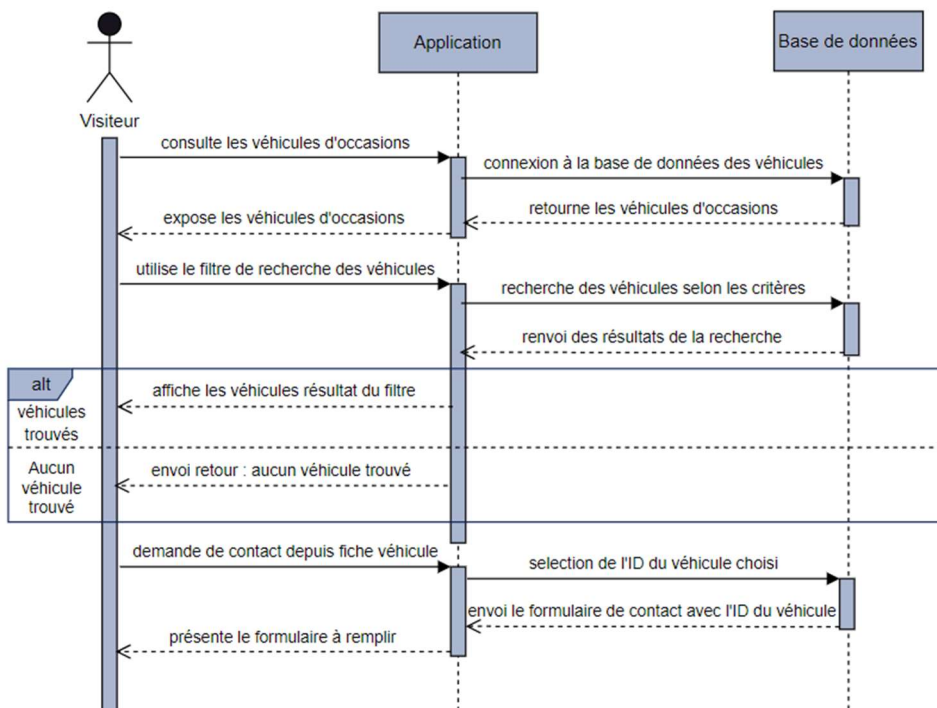


Diagramme de séquence - Garage V. PARROT

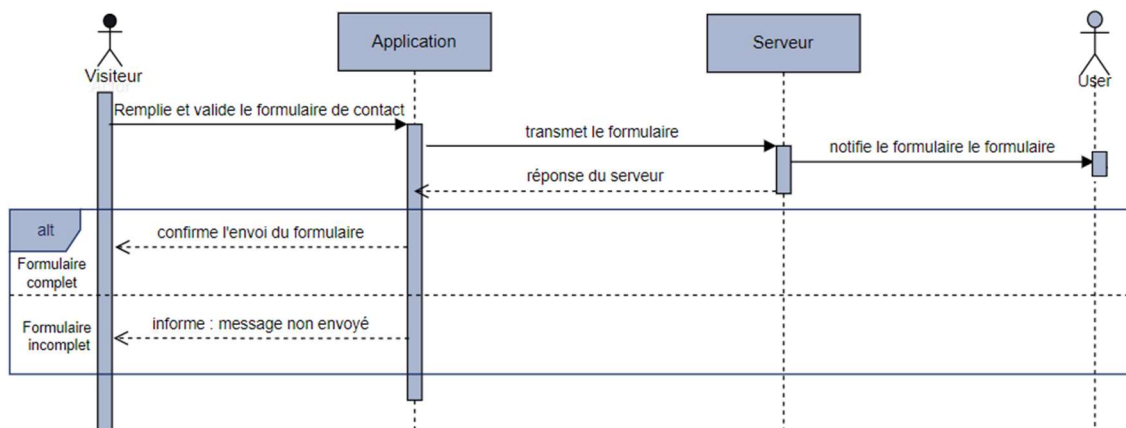
US1. Se connecter



US4. Exposer les voitures d'occasion, US5. Filtrer la liste des véhicules



US6. Permettre de contact l'atelier



US7. Recueillir les témoignages des clients

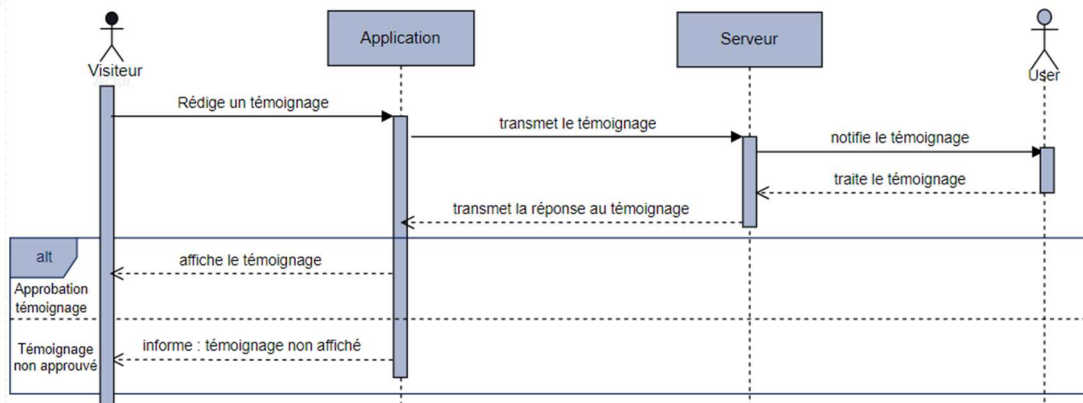
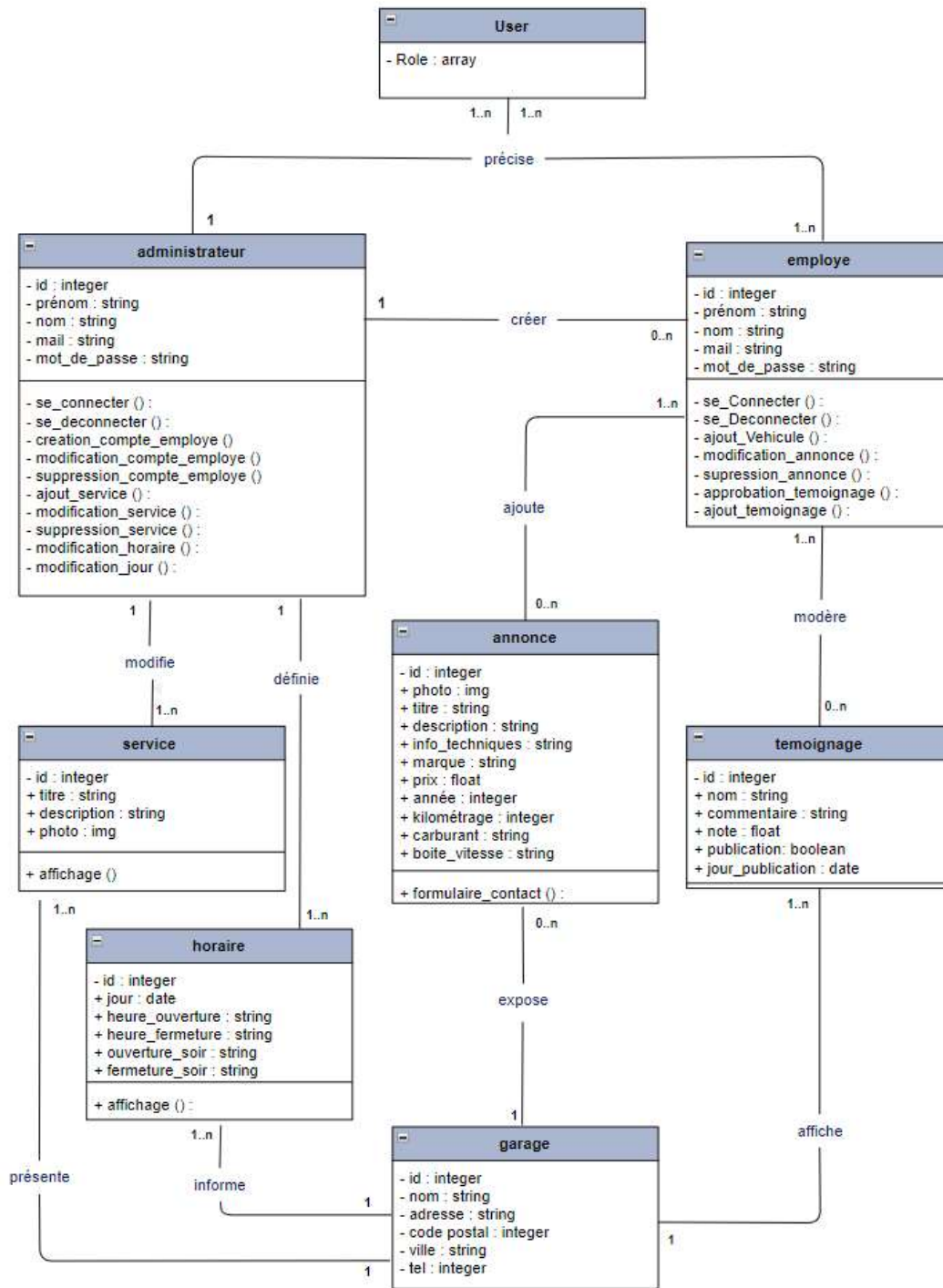


Diagramme de classe - Garage V. PARROT



MESURES DE SECURITE

Le framework Symfony de PHP propose le « bundle Security » qui intègre un ensemble de fonctionnalité permettant la gestion de la sécurité de l'application. Nous les exploiterons en suivant également les recommandations de la CNIL (Commission Nationale de l'Informatique et des Libertés) ainsi que les bonnes pratiques en matière de sécurité.

L'authentification

- Renforcement de la politique de sécurité lors de la création de mot de passe par l'utilisateur. Conformément à l'une des recommandations de la CNIL en matière de protection des mots de passe, nous allons soumettre nos utilisateurs à l'utilisation d'un mot de passe d'un minimum de 12 caractères comprenant des majuscules, des minuscules, des chiffres et des caractères spéciaux. Dans notre cas, ils ne seront visibles que dans la documentation.
- De plus, avec un *Entity Listener*, les mots de passes de tous les utilisateurs de l'application sont systématiquement et automatiquement hashés, à la fois dans l'application et dans la base de données. Notre administrateur fournira donc le mot de passe à ses utilisateurs par un biais en dehors de notre application afin de les maintenir secrets.

Protection contre les attaques de forces brutes

- Cette attaque consiste à tenter de se connecter avec acharnement à l'application en cherchant à devenir le mot de passe. L'utilisateur utilisera successivement des mots de passe
- La première manière de s'en protéger et, comme mentionné ci-dessus, d'imposer des mots de passe long et complexe.
- La deuxième manière consiste tout simplement à limiter dans le temps le nombre de tentative de connexion grâce au composant Rate Limiter. Ainsi on peut configurer le nombre de tentative : au bout de trois essais sans succès, l'utilisateur ne pourra plus tenter de se connecter pendant 30 minutes. C'est une bonne manière de démotiver les utilisateurs malveillants à retenter.
Cependant, dans le cadre de notre application, nous n'irons pas plus loin dans la « sanction » car il est tout à fait possible qu'un employé du garage ou M. Parrot puisse malheureusement commettre des erreurs de saisies. Ainsi, ils pourront retenter leur connexion ultérieurement.

Les autorisations

- La gestion des droits d'utilisateur sont directement assurés par un seul administrateur qui définit le rôle dudit utilisateur à chaque création de compte.
- Uniquement l'administrateur possède l'accès à toutes les pages de l'application. Les autres utilisateurs sont soumis à une restriction de certaines pages et certaines fonctionnalités de l'application grâce aux fonctions de permissions proposées par EasyAdmin.

Protection contre les injections SQL

- Tout d'abord le Bundle Validator rend obligatoire la saisie de données par un utilisateurs grâce au système de « contraintes ». En plus, il permet aussi de soumettre les saisies à certaines contraintes pour s'assurer que ces dernières soient correctes.
- Ensuite, l'ORM Doctrine, utilisé pour interagir avec la base de données propose des mesures de sécurité pour protéger l'application des injections SQL : il utilise pour cela une identification précise des paramètres préparés dans les déclarations SQL.

Protection contre le Cross Site Scripting (XSS)

- L'utilisation du moteur de template TWIG permet de se prémunir contre le Cross Site Scripting (XSS). Cette manœuvre consiste à injecter du code javascript malveillant dans une page web par le biais d'une saisie d'utilisateur. Les conséquences peuvent être grave sur l'application mais aussi entraîner des répercussions sur d'autres utilisateurs.
- C'est notamment grâce à sa syntaxe entre double accolades, qui permet l'échappement des données que TWIG sert de bouclier contre les XSS. Autrement dit, les caractères spéciaux sont convertis en entités HTML dans le navigateur et non pas en code javascript.

Les formulaires

- Le CSRF (Cross Site Request Forgery) permet à un utilisateur malveillant de faire en sorte qu'un utilisateur de l'application soumettent des données à son insu en utilisant le code source du formulaire de l'application. Même si, par exemple notre formulaire « Témoignage » est soumis à une vérification avant publication de M. PARROT et de son équipe, il est important de se prémunir contre ce genre de manœuvre.

Pour s'en prémunir, on va ajouter dans nos formulaires des valeurs secrètes qui prendront la forme « type = 'hidden' ».

Dans son volet Security, Symfony ajoute automatiquement une protection contre le CSRF avec l'implémentation d'un Token lors de la validation d'un formulaire.

Le HTTPS (HyperText Transfer Protocol Secure)

Le framework Symfony inclus le HTTPS dans son budle Security :

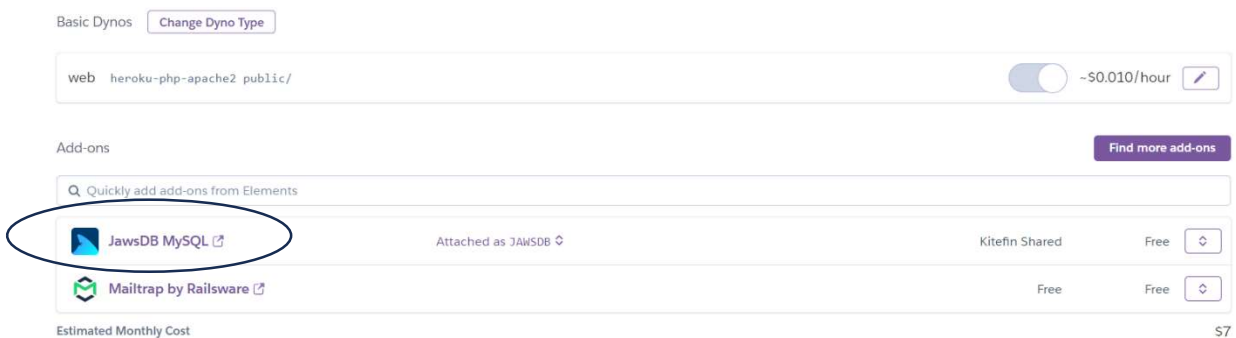
- Le HTTPS garantit la non-modification des données. Cette intégrité des données est importante dans notre application dans la mesure où les utilisateurs sont amenés à saisir des données personnelles dans le formulaire de contact.
- Le HTTPS permet aussi de protéger la confidentialité des données personnelles. Ces dernières restent confidentielles pendant leur transfert entre le client et le serveur et protège ainsi autant les utilisateurs que les membres de l'équipe PARROT qui utilise l'authentification.
- De plus, le HTTPS assure l'authenticité du site visité ou encore un chiffrement des données pour empêcher qu'une personne malveillante les interceptant puisse les utiliser

DEPLOIEMENT SUR HEROKU

L'application a été déployé sur Heroku à l'adresse : <https://fastandparrot-518b85cf506f.herokuapp.com/>. Pour la base de données, j'ai choisi JawsDB MySQL et pour la boîte mail, j'ai choisi MailTrap.

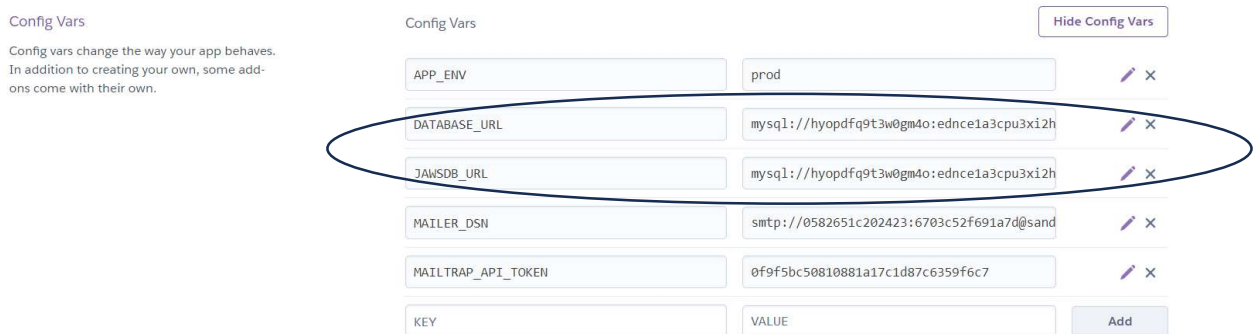
Installation et configuration de la base de données JAWSDB MySQL

Premièrement, on choisit la base de données dans les Add-ons :



Mon choix s'est porté sur cette base de données pour sa facilité de gestion, son support technique et sa documentation bien fournie. Cette base de données est également réputée pour sa performance et sa gratuité.

Pour la connecter à son application et réussir ensuite la migration lors du déploiement, les variables d'environnement doivent être bien configurées :



La documentation de JawsDB MySQL sur le site d'Heroku (<https://devcenter.heroku.com/articles/jawsdb>) nous détaille la procédure à suivre pour générer intégralement sa base de données locale, dans un fichier backup.sql, puis la faire migrer vers la base de données d'Heroku

Manual Backups and Migrations with mysqldump

`mysqldump` is a powerful client utility packaged with the MySQL clients available on the [MySQL downloads page](#). With this utility, users can backup their database into flat files as well as restore those backups to any MySQL or compatible server of their choice.

In order to complete either operation, the connection credentials and information of the relevant server(s) must be known in advance.

Connection strings provided by JawsDB contain all of the relevant info in the following format:

```
mysql://username:password@hostname:port/database
```

`username`, `password`, `hostname`, `port`, and `database` will be referenced in the following sections.

The `mysqldump` tutorials below are simplistic and general. The full range of options and configurations that `mysqldump` supports can be seen [here](#) and in the tool's own `man` pages.

Manual Backups

The following command will backup the specified database into a local file called `backup.sql`

```
mysqldump --no-tablespaces -h hostname -u username -ppassword database > backup.sql
```

Note There is no space between the `-p` flag and `password`

If using MySQL v.8, you may need to add `--column-statistics=0` to your `mysqldump` command

Manual Migrations

Backup files created with `mysqldump` can be loaded onto a target server the way any SQL file would be loaded. Below is a way to do this with the `mysql` command-line utility.

```
mysql -h hostname -u username -ppassword database < backup.sql
```

Note again how there is no space between the `-p` flag and `password`

D'autres requêtes peuvent être ajoutées manuellement ensuite, notamment la création d'un User :

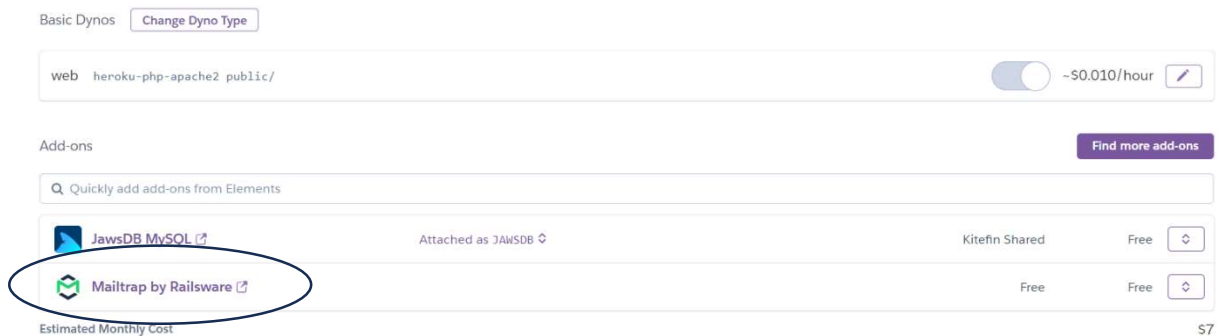
```
INSERT INTO `user`  
(`id`, `email`, `roles`, `password`, `nom`, `prenom`, `category_id`)  
VALUES  
(NULL, 'harrywinsley@vparrot.com', '[]',  
'$2y$13$Qai4nr6WhcjzOESrD8NAd.AAZqWeUA41X61gQMFPMSIKsD1qwjoEu', 'Winsley',  
'Harry', NULL);
```

A noter : le mot de passe hashé est généré grâce à la ligne de commande : `symfony console security:hash-password`

Configuration de la boîte mail MAILTRAP.

L'avantage de cette interface est sa facile prise en main et la possibilité qu'elle offre d'utiliser des boîtes mail en local et en production. Plusieurs options sont proposées à différents coûts, mais en développement et production, les formules gratuites répondent parfaitement à nos besoins dans le cadre de cette application.

Pour commencer, nous installons l'add-on :



Puis nous complétons les variables d'environnements nécessaires à l'aide des informations données sur la messagerie, à savoir le MAILER_DSN et l'API Token :

Mail

Total messages sent: 0

SMTP Settings Email Address Auto Forward Manual Forward Access Rights

SMTP / POP3 ⓘ Reset Credentials 🔑

Use these settings to send messages directly from your email client or mail transfer agent.

⚠ Don't disclose your username or password as this may result in your inbox getting filled up with spam.

Show Credentials ▾

Integrations ⓘ

Symfony 5+ ⌵

Symfony uses Symfony Mailer to send emails. You can find more information on how to send email on [Symfony's website](#).

To get started you need to modify .env file in your project directory and set MAILER_DSN value:

MAILER_DSN=smtp://0582651c202423:*****1a7d@sandbox.smtp.mailtrap.io:2525

Copy

User settings



API v1 Authentication

API Token: 0f9f5bc50810881a'

Copy

JWT Token: eyJhbGciOiJIUzUxM







Copy

[Reset API/JWT Token](#)

Nous les reportons ensuite dans nos variables Heroku :

Config Vars

Hide Config Vars

APP_ENV	prod	 
DATABASE_URL	mysql://hyopdfq9t3w0gm4o:ednce1a3cpu3xi2h	 
JAWSDB_URL	mysql://hyopdfq9t3w0gm4o:ednce1a3cpu3xi2h	 
MAILER_DSN	smtp://0582651c202423:6703c52f691a7d@sand	 
MAILTRAP_API_TOKEN	0f9f5bc50810881a17c1d87c6359f6c7	 
KEY	VALUE	

Reportez-vous dans le Manuel d'utilisation pour vous connecter à la boîte mail en question et tester l'envoi des mails.