

Fontana Benjamin

Cheffe de projet : HARDEGGER Cindy

Expert 1 : MONTEMAYOR Ernesto

Expert 2 : BERTINO Yves

Table des matières

[Analyse préliminaire 4](#_Toc136527529)

[Introduction 4](#_Toc136527530)

[Objectifs 4](#_Toc136527531)

[Planification initiale 4](#_Toc136527532)

[Analyse / Conception 5](#_Toc136527533)

[Concept 5](#_Toc136527534)

[Base de données 5](#_Toc136527535)

[Cate du site 7](#_Toc136527536)

[Wireframes 7](#_Toc136527537)

[Stratégie de test 7](#_Toc136527538)

[Risques techniques 7](#_Toc136527539)

[Planification 8](#_Toc136527540)

[Méthode de gestion de projet 8](#_Toc136527541)

[Planification définitive 8](#_Toc136527542)

[Dossier de conception 9](#_Toc136527543)

[Hardware 9](#_Toc136527544)

[Software 9](#_Toc136527545)

[Maquettes 9](#_Toc136527546)

[Hébergement 9](#_Toc136527547)

[Mise à jour de l’application 9](#_Toc136527548)

[Base de données 10](#_Toc136527549)

[Diagramme de flux 13](#_Toc136527550)

[Architecture de l’application 13](#_Toc136527551)

[Réalisation 14](#_Toc136527552)

[Dossier de réalisation 14](#_Toc136527553)

[Répertoire de l’application 14](#_Toc136527554)

[Technologies et versions utilisée 14](#_Toc136527555)

[Librairies 14](#_Toc136527556)

[Matériel hardware utilisé 14](#_Toc136527557)

[Version d’aeromodel 14](#_Toc136527558)

[Librairie et données personnalisée 14](#_Toc136527559)

[Implémentation des QR codes 15](#_Toc136527560)

[Implémentation de la modification de mot de passe 15](#_Toc136527561)

[Gestion des erreurs de saisie 16](#_Toc136527562)

[Implémentation de la Suppression 16](#_Toc136527563)

[Responsive 16](#_Toc136527564)

[Description des tests effectués 17](#_Toc136527565)

[Erreurs restantes 29](#_Toc136527566)

[Liste des documents fournis 29](#_Toc136527567)

[Conclusions 29](#_Toc136527568)

[Objectifs 29](#_Toc136527569)

[Atteints 29](#_Toc136527570)

[non-atteints 29](#_Toc136527571)

[Points positifs 29](#_Toc136527572)

[Points négatifs 29](#_Toc136527573)

[Difficultés 30](#_Toc136527574)

[Suites 30](#_Toc136527575)

[Glossaire 31](#_Toc136527576)

[Annexes 32](#_Toc136527577)

[Résumé du rapport du TPI 32](#_Toc136527578)

[Sources – Bibliographie 32](#_Toc136527579)

[Journal de travail 32](#_Toc136527580)

[Manuel d'Installation 32](#_Toc136527581)

[Manuel d'Utilisation 32](#_Toc136527582)

[Archives du projet 32](#_Toc136527583)

# Analyse préliminaire

## Introduction

Le but de ce projet consiste en la réalisation d’une application web pour smartphone et pc dédié aux passionnés d’aéromodéliste qui permet à terme à chaque utilisateur de répertorier leurs modèles réduits télécommandés et pour chacun d’eux d’avoir un suivi des vols effectués ainsi que de la gestion des crashs, des réparations et du cycle de vie des accumulateurs.

Il sera effectué dans le cadre de mes études en tant que projet final de TPI. Il m’apportera beaucoup d’expérience dans la gestion de projet ainsi que sur les technologies choisies.

## Objectifs

* L’utilisateur peut supprimer son compte via la page de profil.
* L’utilisateur peut modifier son mot de passe via la page de profil.
* L’utilisateur peut modifier les données de ses modèles réduits.
* L’utilisateur peut supprimer un de ses modèles réduits.
* L’utilisateur peut ajouter un accumulateur et l’associer avec un modèle réduit.
* L’utilisateur peut ajouter un moteur et l’associer avec un modèle réduit.
* Pour chaque modèle, un QR-Code est généré. Le QR-Code pourra être imprimé et être utilisé pour une identification unique et ainsi offrir un partage simplifié avec d’autres utilisateurs. Le QR-Code permet de rediriger directement sur le modèle.
* L’application web offre une interface responsive adaptée au smartphone.

## Planification initiale

Le projet sera réalisé dans la période du 2 mai 2023 au 2 juin 2023.

Vous trouverez le diagramme de Gantt de la planification initiale en annexe à la page

Durant cette période, le projet sera réparti comme ci-après :

# Analyse / Conception

## Concept

### Base de données

#### Modèle conceptuel de données

A screenshot of a computer

Description automatically generated with medium confidenceVous pouvez constater que dans ce MCD, certaines entités n’ont pas de contraintes d’unicité tels que les modèles, les batteries et les moteurs. Car dans ces cas, l’unicité se fera avec l’id de chaque entrée. Vous pouvez aussi constater que les batteries et les moteurs sont tous les deux liés à l’utilisateur avec un lien 0,n. Car, il y’a des batteries et des moteurs par défaut mais l’utilisateur peut aussi en créer.

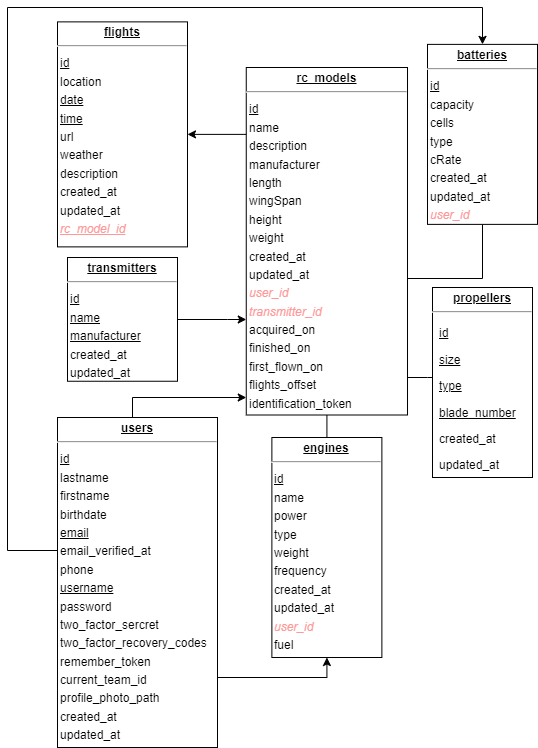
#### Modèle logique de données

Vous pouvez trouver ci-dessous le MLD. Il traduit le MCD ci-dessus en ajoutant donc les contraintes d’unicité sur les id des batteries, moteurs et modèles. Il y’a aussi deux champs qui ont été ajouté à toutes les tables : created\_at, updated\_at utilisé dans par le code pour garder une trace des actions effectuées.

Légende

*Clés étrangères*

Index d’unicité ou nom de la\_table



### A picture containing black, darkness Description automatically generatedCate du site

### Wireframes

Les wireframes sont basés sur le design fait par [Isaac Anyanwu](https://www.figma.com/@iceycreative) « [Full Mobile wallet UI](https://www.figma.com/community/file/1106886829064472050)».

Vous les retrouverez tous en annexe de la page ASD à la page ASD ou en accédant au projet figma se trouvant dans le répertoire ./documents/Design/AeromodelDesign.fig du repository GitHub.

## Stratégie de test

Les tests qui seront effectués seront manuels. Le backend sera testé via le frontend et le frontend sera testé manuellement. Les tests seront exhaustifs pour couvrir tous les cas de figures.

Pour les effectuer, des données seront générées. Ces données sont partiellement réelles pour les modèles réduits, les télécommandes et les moteurs. Toutes les autres sont générée avec la librairie faker pour l’utilisateur et avec mockaroo pour toutes les autres.

Je serais l’unique testeur professionnel lors de ce projet. Je prévois de faire tester l’accessibilité de l’application à un modéliste qui accessoirement est mon père.

## Risques techniques

Je n’ai jamais pratiqué l’impression via une application web et n’ai jamais généré de QR codes. C’est donc pour moi les plus grandes difficultés que je peux rencontrer lors de ce projet.

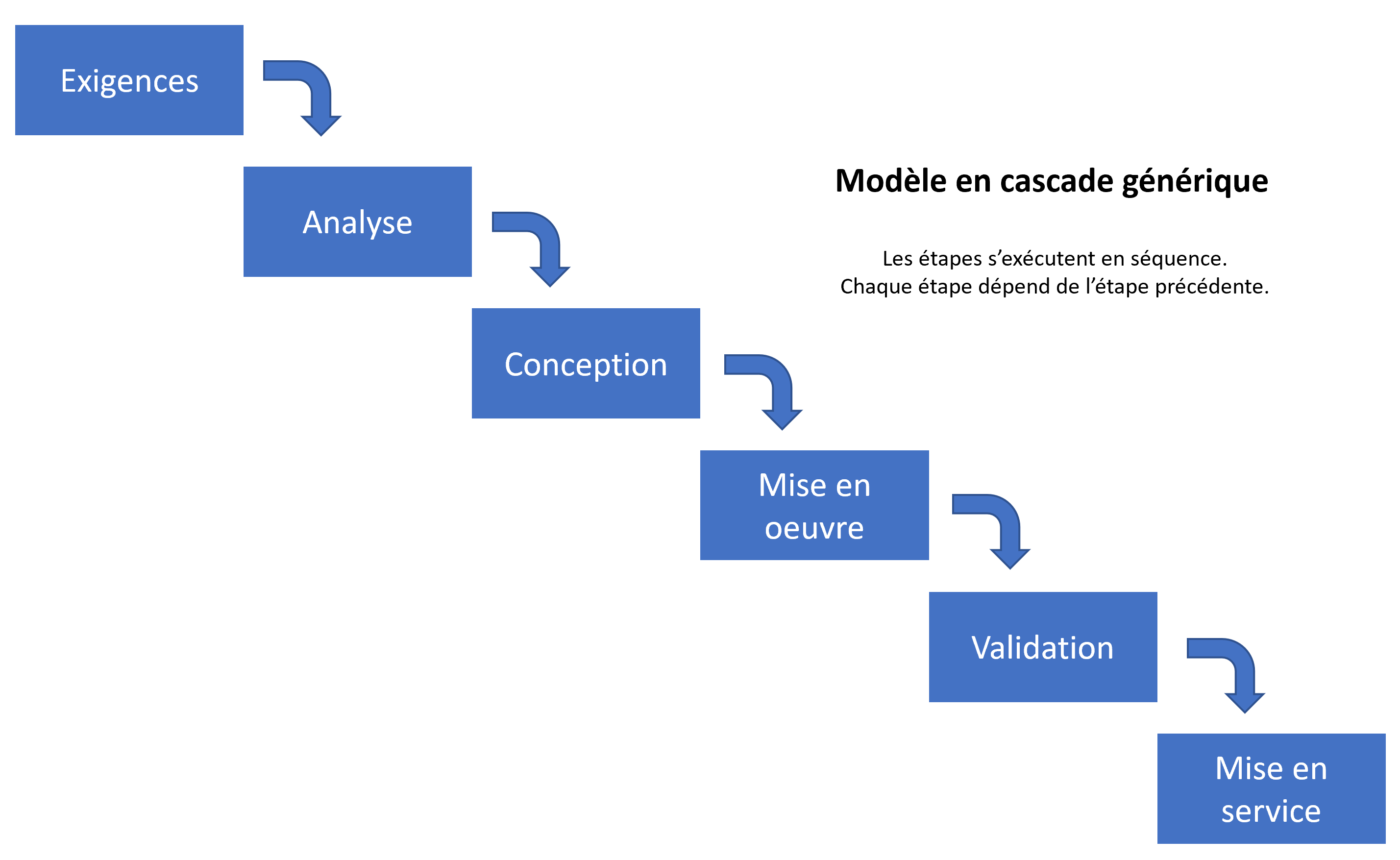
Une autre difficulté sur laquelle je pourrais me heurter durant le développement serait due au fait que je continue un projet de pré-TPI qui n’était pas fini. Il se peut donc que certaines erreurs me ralentissent.

Pour ne pas me laisser dépasser par ces risques, j’ai priorisé les fonctionnalités les plus importantes au début puis, me suis laissé assez de temps pour les développer.

J’ai aussi recherché de bonnes documentations sur l’utilisation des QR code pour déjà apprivoiser leur utilisation.

## Planification

### Méthode de gestion de projet

Ce projet se fera sous la méthode Waterfall (Cascade). Toutes les tâches seront planifiées dans quatre sprints. Elles seront écrites sous la forme de user stories : Qui, Quoi, Pourquoi.

### Planification définitive

La planification finale comporte quelques changements dû à l’ajout de tâche plus précise et de l’ajout de la mise en production de l’application.

Vous pouvez retrouver la planification finale en annexe à la page

## Dossier de conception

### Hardware

* Ordinateur type CPNV
* Smartphone

### Software

* Windows
* WSL sous Kali Linux
* Visual studio code
* Docker
* Figma
* GitKraken (Git)
* Notion
* Draw.io
* MySQL Workbench (Visualisation DB)
* Google Chrome/Samsung internet
* Word
* Excel

# 

### Maquettes

Les maquettes ont été adaptée du design fait par [Isaac Anyanwu](https://www.figma.com/@iceycreative) « [Full Mobile wallet UI](https://www.figma.com/community/file/1106886829064472050)» L’UI « [Full Mobile wallet UI](https://www.figma.com/community/file/1106886829064472050) » est sous la licence [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/).

Vous retrouverez toutes les maquettes en annexe de la page ASD à ASD ou en accédant au projet figma se trouvant dans le répertoire ./documents/Design/AeromodelDesign.fig du repository GitHub.

### Hébergement

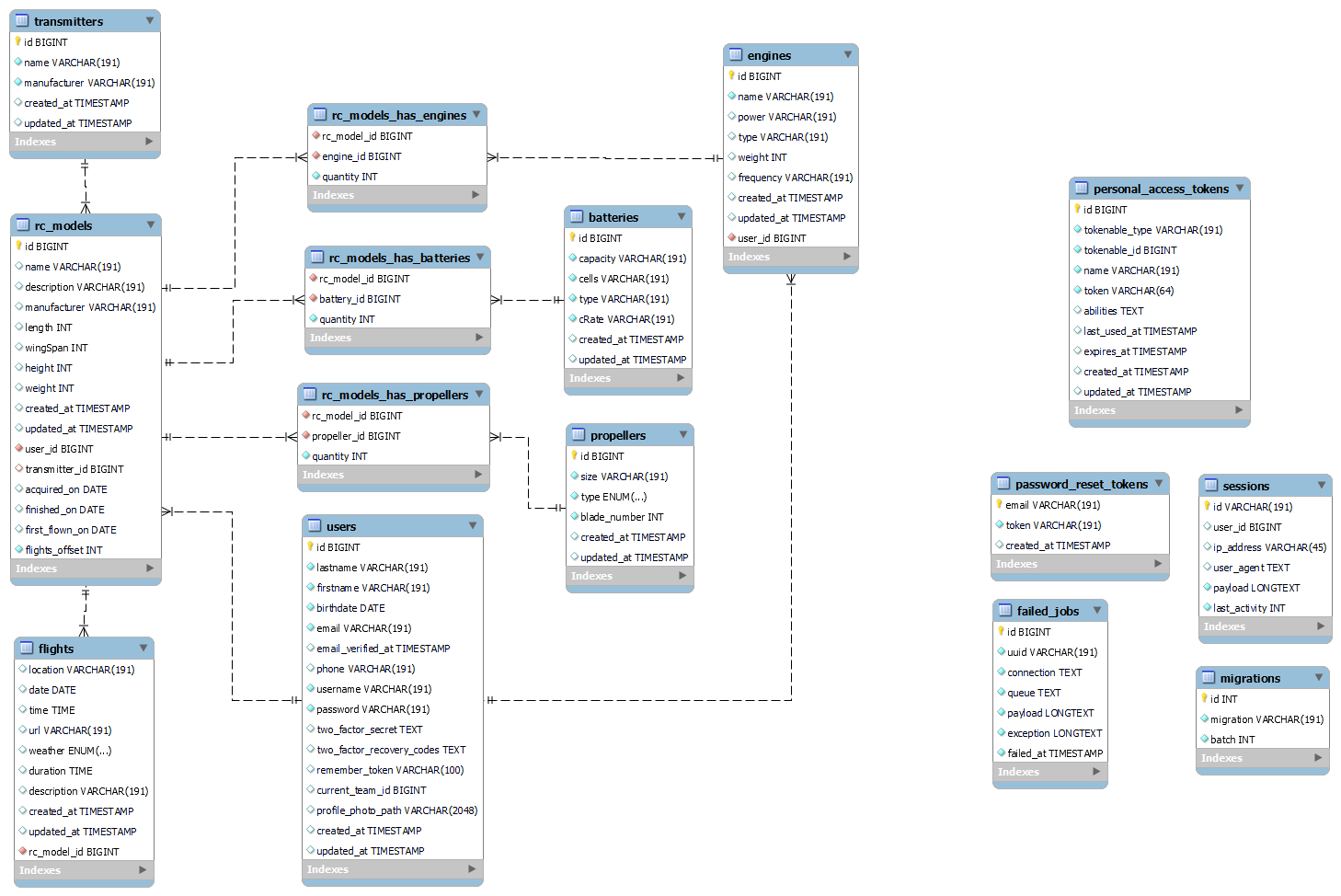
Le site sera hébergé sur un serveur SwissCenter sous le nom de domaine « aeromodel.mycpnv.ch ».

### Mise à jour de l’application

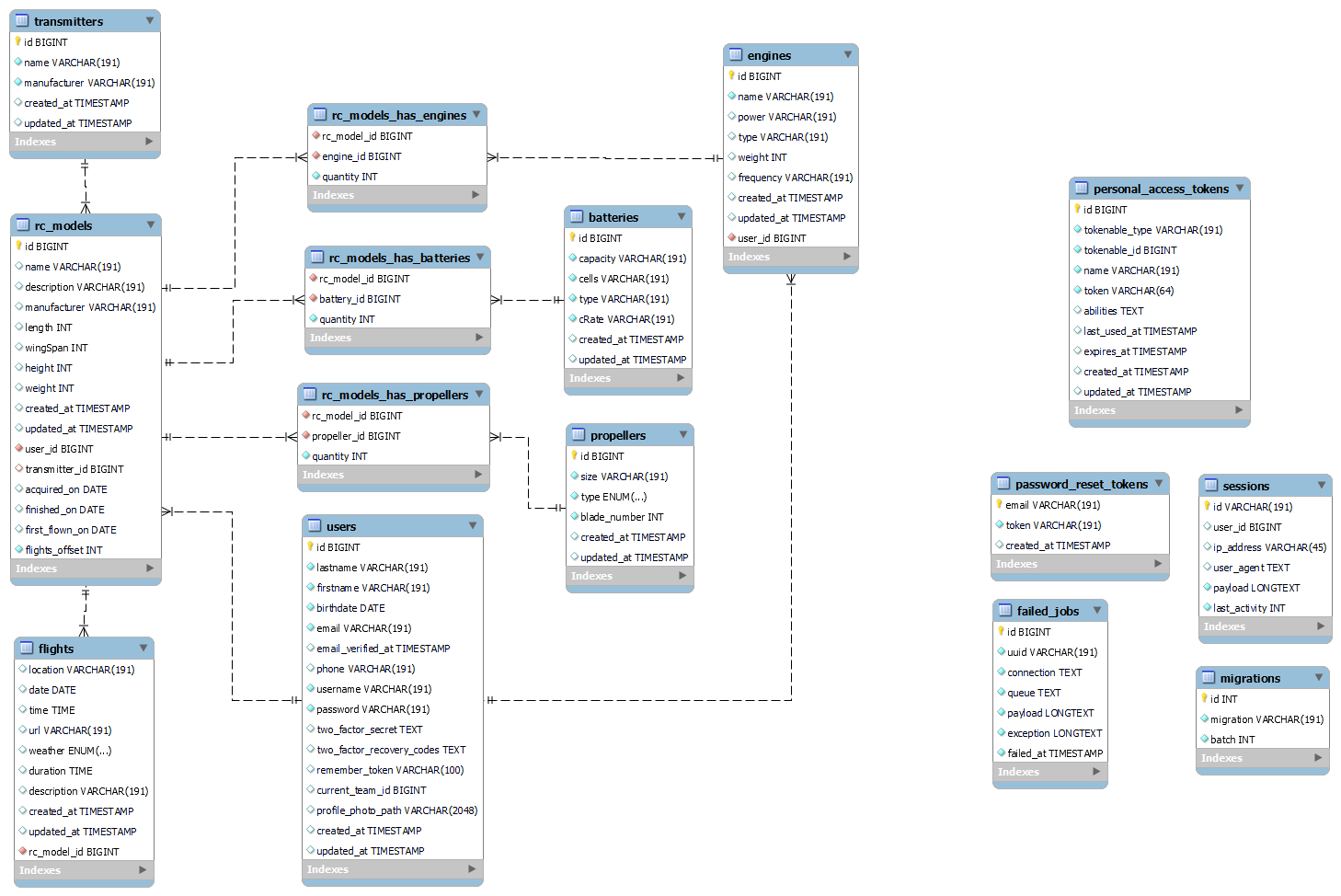
Le serveur fera pourra effectuer un pull du repository GitHub pour se mettre à jour. Tout se fera manuellement pour ne pas détruire le site de production pas mégarde.

### Base de données

#### Modèle Physique de données

Voici le MPD de l’application. Vous trouverez ci-dessous sur la figure ci-dessous les tables d’aeromodel.

Voici ci-dessous les table de laravel et fortify qui n’ont aucun lien direct avec les tables d’aeromodel.



#### Requête de données

L’application utilise l’ORM « Eloquent » de Laravel pour faire les requêtes à la base de données. Il n’y a donc aucune requêtes SQL exhaustive mais seulement des manipulations d’objet et de requête sur l’objet de type mise à jour, suppression, etc…

A screen shot of a computer

Description automatically generated with low confidence

Figure - Récupération des modèles de l'utilisateur connecté

A screen shot of a computer program

Description automatically generated with medium confidence

Figure - Création d'un modèle dans la base de données

## Diagramme de flux

## Architecture de l’application

L’application est basée sur un starter kit Jetstream. Il y’a donc plusieurs répertoires importants et principaux. Mais l’architecture MVC peut être grossièrement retrouvée dans l’architecture Jetstream.

|  |  |
| --- | --- |
| MVC | Jetstream |
| Model | ./app/Models |
| View | ./resources/js/Pages |
| Controller | ./app/http/Controllers |

Puisque le frontend utilise Vue comme framework principal, il y’a aussi d’autres répertoire important pour les composants. Vous pouvez retrouver un descriptif plus précis en annexe à la page ASD

# Réalisation

## Dossier de réalisation

### Répertoire de l’application

Vous pouvez accéder au repository Github de l’application au lien suivant : <https://github.com/R0kkxSynetique/TPI>

Il contient l’application à la racine du repository et les documents dans le répertoire ./documents.  
Vous pouvez retrouver en annexe la liste de tous les fichiers utilisés accompagné d’une brève description de leur utilité.

### Technologies et versions utilisée

* Windows 11
* WSL 2
* Kali linux 2022.4
* PHP 8.2.3
* Laravel 10
* JetStream 3 avec Inertia
* Tailwind CSS 0.5.2
* Vue 3
* Vite 4

### Librairies

* Moment JS
* HeadlessUI
* HeroIcons
* Axios
* Vue-toastification
* simple-qrcode
* PrintJS

### Matériel hardware utilisé

* Laptop Lenovo ideapad gaming 3
* Samsung Galaxy S21 Ultra

### Version d’aeromodel

Aeromodel est actuellement passé à la version 0.2.

### Librairie et données personnalisée

La librairie image.ts utilisée dans ce projet provient de Kénan Augsburger.

Les images de tests pour les modèles dans ce projet proviennent de Alain Fontana.

### Implémentation des QR codes

Pour l’implémentation des QR codes, il a fallu importer de nouvelles librairies. Dans ce projet, j’ai importé simple-qrcode de SimpleSoftwareIO et PrintJS de crabbly. Cette première librairie me permet de créer des QR codes un peu personnalisé en php.

Un contrôleur destiné uniquement aux QR code à été créer. Dans ce contrôleur, il y’a deux fonctions. Une qui retourne un download du svg du QR code ce qui permet à la librairie d’impression de montrer le QR code dans la prévisualisation d’impression.

L’autre fonction génère le token d’identification et le QR code. Le token est un hash md5 composé de plusieurs variables.

Il y’a un sel, l’id de l’utilisateur connecté et l’id du modèle lié.

A screen shot of a computer

Description automatically generated with low confidence

Figure - Création du token d'identification

Lors de l’accès au modèle via un QR code, la route utilisée requière le token du modèle cible pour autoriser l’accès aux données. Si celui-ci n’est pas identique à celui de la base de données, l’affichage du modèle est refusé. En tant qu’invité, l’utilisateur n’a aucunes actions possibles sur le modèle hormis le consulter.

### Implémentation de la modification de mot de passe

L’implémentation de la modification de mot de passe s’est faite via une action fortify. Par défaut, lors de la création d’un projet Laravel avec JetStream, fortify possède déjà des actions standard. Malheureusement, l’action basique ne convenait pas à mes besoins. J’ai l’ai donc mise à jour avec un validateur comprenant mes mots de passe et, ai ajouté le dossier `lang` et les fichiers de traduction pour retourner des erreurs en français. Une fois la validation passée, le nouveau mot de passe est haché avec Bcrypt puis sauvegardé dans la base de données. Puisque j’utilise le middleware de Sanctum sur les routes, l’utilisateur est déconnecté, le forçant à entrer son nouveau mot de passe pour continuer à utiliser l’application.

### Gestion des erreurs de saisie

La gestion est principalement faite en frontend avec les erreurs HTML. Cela permet de restreindre les utilisateurs standard. Les seuls moments où une validation est faite en backend est lors de la gestion des utilisateurs ou lors de la création d’un modèle pour confirmer que l’image est bien présente.  
Les erreurs des validations sont affichée dans des toasts. A red rectangle with white text

Description automatically generated with medium confidence

### Implémentation de la Suppression

La suppression est uniquement du « hard delete ». Les données sont directement supprimées et ne peuvent pas être récupérée.

Si un utilisateur supprime son compte, alors tous ses modèles le seront aussi. En revanche, les moteurs et les batteries qu’il a créé seront mis à disposition des autres utilisateurs.

### Responsive

A screenshot of a phone

Description automatically generated with low confidenceLe responsive implémenté est adapté du mobile au pc. Puisque tailwindCSS est « mobile-first » les classe sont en premier lieu ciblée pour les mobiles puis doivent être adaptée pour pc.

A screenshot of a phone

Description automatically generated with low confidenceL’application est donc responsive pour ces deux plateformes. Sur smartphone, la major partie du temps les éléments prendront chacun la largeur de l’écran(ex. les modèles réduits sont affichés un par un). Tandis que sur ordinateur, l’affichage se fait avec plusieurs éléments par ligne.

Figure - Affichage de l'accueil sur desktop

Figure - Affichage de l'accueil sur smartphone

## 

## Description des tests effectués

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test effectué | Résultat attendu | Moyens | Description | Résultat |
| Création d’un compte utilisateur avec toutes les informations demandée et correctes | L’utilisateur est automatiquement authentifié et est redirigé vers la page d’accueil. | URL: <http://localhost/register>  Pseudo: JDoe  Nom: Doe  Prénom : John  Date de naissance: 01.01.2001  Adresse email: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01 | Le formulaire est envoyé et nous sommes redirigés sur la page d’accueil en étant connecté. | OK |
| Création d’un compte utilisateur avec des informations manquantes | L’utilisateur peut voir une erreur et peut compléter les champs requis pour se créer son compte. | URL: <http://localhost/register>  Pseudo: -  Nom: Aubé  Prénom: Alice  Date de naissance: -  Adresse email: [alice.aube@aero.ch](mailto:alice.aube@aero.ch) Mot de passe : aaube15 | Les champs ont la validation HTML qui demande que le champ soit rempli. | OK |
| Création d’un compte utilisateur avec un email déjà existant. | L’utilisateur peut voir une erreur lui indiquant que l’email est déjà utilisé | URL: <http://localhost/register>  Pseudo: JDupuis  Nom: Dupuis  Prénom : Julien  Date de naissance: 23.12.1989  Adresse email: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdupuis23 | Un toast est affiché en indiquant que l’email est déjà utilisé. | OK |
| Création d’un compte utilisateur avec un nom d’utilisateur déjà existant. | L’utilisateur peut voir une erreur lui indiquant que le nom d’utilisateur est déjà utilisé | URL: <http://localhost/register>  Pseudo: JDoe  Nom: Allaire  Prénom : Romane  Date de naissance: 20.10.1976  Adresse email: [romane.allaire@aero.ch](mailto:romane.allaire@aero.ch) Mot de passe : rallaire20 | Un toast est affiché en indiquant que le nom d’utilisateur est déjà utilisé. | OK |
| Connexion avec un compte utilisateur existant et avec les bonnes informations | L’utilisateur peut se connecter avec un compte existant et est redirigé sur la page d’accueil une fois le formulaire envoyé. | URL: <http://localhost/login>  Adresse email: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01 | En entrant l’email d’un compte crée dans la base de données l’utilisateur est authentifié et est redirigé sur la page d’accueil. | OK |
| Connexion avec un compte utilisateur non existant | Une erreur est affichée pour informer l’utilisateur que le compte n’existe pas. | URL: <http://localhost/login>  Adresse email: [alice.aube@aero.ch](mailto:alice.aube@aero.ch)  Mot de passe : aaube15 | Un toast est affiché pour indiquer qu’il y’a une erreur. | OK |
| Connexion avec un compte utilisateur existant mais avec un mauvais mot de passe | Une erreur est affichée pour informer l’utilisateur que la combinaison email et mot de passe n’est pas correcte | URL: <http://localhost/login>  Adresse email: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : aaube15 | Un toast est affiché pour indiquer qu’il y’a une erreur. | OK |
| Déconnexion depuis une page de l’application | L’utilisateur connecté peut ouvrir le menu depuis lequel il peut se déconnecter puis être redirigé sur la page de login. | Adresse email: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01 | L’utilisateur est bien déconnecté et est redirigé sur la page de connexion. | OK |
| Accès au formulaire de création d’un modèle | L’utilisateur peut appuyer sur le bouton « créer » qui lui est montré pour accéder à la page de création d’un modèle | URL: <http://localhost>  Adresse email: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01 | En appuyant sur le bouton, l’utilisateur est redirigé sur la page de création d’un modèle. | OK |
| Création d’un modèle | L’utilisateur peut créer un modèle en remplissant tous les champs du formulaire | Image : [a6m zero](https://scontent.fqls2-1.fna.fbcdn.net/v/t39.30808-6/345648776_686793950121121_3232030770404612174_n.jpg?_nc_cat=103&ccb=1-7&_nc_sid=8bfeb9&_nc_ohc=2AMLa18hZcsAX8qmtc4&_nc_ht=scontent.fqls2-1.fna&oh=00_AfA7In9351JzZOwID-hWIqvIWZ8QHaE2LLFvNYSoblybTg&oe=647455D5)  Nom du modèle : A6M zero  Constructeur : Meister scale  Description : Avion de chasse japonais  Envergure : 3008  Poids : 16800  Longueur : 2300  Hauteur : 800  Moteur : fiala B2 120 cc  Hélice : 27x10 bois  Batteries : 2s 2800mAh 25c LiPo  Acquis le : 08.02.2017  Fini le: 15.03.2018 Mis en service le : 22.03.2018 Radiocommande : Futaba t16 sz vols : 2 | Le formulaire est envoyé et nous somme redirigés sur la page d’accueil. | OK |
| Affichage des modèles | L’utilisateur peut son/ses modèle(s) sur la page d’accueil | URL : <http://localhost> | Le modèle qui vient d’être crée est affiché sur la page d’accueil. | OK |
| Accès à la création d’un deuxième modèle | L’utilisateur est capable d’accéder au formulaire de création d’un modèle depuis la page d’accueil en appuyant sur le bouton créer. | URL : <http://localhost> | Nous sommes bien redirigés sur la page de création d’un modèle. | OK |
| Création d’un modèle avec les données minimum | L’utilisateur peut créer un modèle avec uniquement une image et un nom | URL : <http://localhost/rc-models/create>  Image : ./tests/images/zero.jpg  Nom du modèle : A6M zero | Le formulaire est bien envoyé et nous sommes redirigés sur la page d’accueil. | OK |
| Affichage d’un modèle avec le minimum de données | L’utilisateur peut voir un modèle qui possède uniquement une image et un nom sur la page d’accueil. Si l’utilisateur supprime le nom du modèle, il est aussi affichable. |  | Le modèle est bel et bien affiché sur la page d’accueil. | OK |
| Accès aux détails d’un modèle sélectionné | L’utilisateur peut voir les détails du modèle sur lequel il clique lorsqu’il est sur l’accueil | Accéder à la page de consultation en appuyant sur le nom, l’image ou les données du modèle en question sur la page d’accueil. | Nous sommes bien redirigés sur la page des détails du modèle sélectionné. | OK |
| Vérification de l’intégrité des données | Les données qui ont été remplie lors de la création du modèle sont toutes visibles et correctes. |  |  | OK |
| Suppression d’un modèle | L’utilisateur est capable de supprimer un modèle. Une confirmation d’action est affichée avant de le supprimer. Il peut supprimer un modèle depuis la page de consultation ou depuis la page d’édition de celui-ci | Appuyez sur le menu ‘’…’’ puis sur l’option « supprimer ». | En supprimant le modèle depuis la page de consultation, une confirmation de suppression est affichée et nous pouvons annuler notre action. Si nous acceptons, nous sommes redirigés sur la page d’accueil et un toast nous confirme que le modèle a été supprimé. Depuis la page d’édition le processus est le même. | OK |
| Annulation de la suppression | Si l’utilisateur appuie sur annuler le modèle n’est pas supprimé |  | En cliquant sur annuler, l’action n’est pas effectuée et le modèle n’est donc pas supprimé. | OK |
| Affichage d’un modèle possédant plusieurs fois le même moteur | L’affichage d’un modèle avec plusieurs fois le même moteur est correcte et nous pouvons voir le nombre total des moteurs. | Utilisateur : [test@aero.ch](mailto:test@aero.ch)  Mot de passe : password  Nom du modèle : de Havilland DH.98 Mosquito | En se connectant avec le compte cité ci-avant, puis en accédant au modèle indiqué, nous pouvons effectivement constater que les moteurs sont bien énumérés. | OK |
| Affichage d’un modèle possédant plusieurs moteurs différents | L’affichage d’un modèle avec plusieurs moteurs différents est correct et nous pouvons voir les modèles de moteur et leur quantité. | Utilisateur : [test@aero.ch](mailto:test@aero.ch)  Mot de passe : password  Nom du modèle : Consolidated B-24 Liberator | En se connectant avec le compte cité ci-avant, puis en accédant au modèle indiqué, nous pouvons effectivement constater que les moteurs sont bien énumérés. | OK |
| Affichage d’un modèle possédant plusieurs fois la même batterie | L’affichage d’un modèle avec plusieurs fois la même batterie est correcte et nous pouvons voir le nombre total des batteries | Utilisateur : [test@aero.ch](mailto:test@aero.ch)  Mot de passe : password  Nom du modèle : Hawker Hurricane | En se connectant avec le compte cité ci-avant, puis en accédant au modèle indiqué, nous pouvons effectivement constater que les batteries sont bien énumérées. | OK |
| Affichage d’un modèle possédant plusieurs batteries différentes | L’affichage d’un modèle avec plusieurs batteries différentes est correct et nous pouvons voir les modèles de batteries et leur quantité. | Utilisateur : [test@aero.ch](mailto:test@aero.ch)  Mot de passe : password  Nom du modèle : Mitsubishi A6M Zero | En se connectant avec le compte cité ci-avant, puis en accédant au modèle indiqué, nous pouvons effectivement constater que les batteries sont bien énumérées. | OK |
| Affichage d’un modèle possédant plusieurs fois la même hélice | L’affichage d’un modèle avec plusieurs fois la même hélice est correcte et nous pouvons voir le nombre total des hélice | Utilisateur : [test@aero.ch](mailto:test@aero.ch)  Mot de passe : password  Nom du modèle : de Havilland DH.98 Mosquito | En se connectant avec le compte cité ci-avant, puis en accédant au modèle indiqué, nous pouvons effectivement constater que les hélices sont bien énumérées. | OK |
| Affichage d’un modèle possédant plusieurs hélices différentes | L’affichage d’un modèle avec plusieurs hélices différentes est correct et nous pouvons voir les modèles d’hélices et leur quantité. | Utilisateur : [test@aero.ch](mailto:test@aero.ch)  Mot de passe : password  Nom du modèle : Consolidated B-24 Liberator | En se connectant avec le compte cité ci-avant, puis en accédant au modèle indiqué, nous pouvons effectivement constater que les hélices sont bien énumérées. | OK |
| Modification d’un modèle existant | Il est possible de modifier un modèle existant depuis la page de consultation du modèle. | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Modèle : A6M zero  Appuyer sur le menu ‘’…’’ puis sur l’option modifier. Changer les valeurs souhaitées puis appuyer sur le bouton « sauver » | Le formulaire est envoyé nous sommes redirigés sur la page de consultation et les données sont modifiées. Un toast nous indique que le modèle à bien été mis à jour. | OK |
| Création d’une batterie lorsque l’utilisateur ne n’en possède pas encore | L’utilisateur est capable de créer une batterie si il ne n’en possède pas encore en cliquant sur un bouton. | URL : <http://localhost/batteries>  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Capacité : 800, Cellules : 3  C-Rate : 25, Type : LiPo  Appuyer sur le bouton « créer » pour accéder au modal de création. | Le modal s’ouvre nous pouvons remplir les champs et sélectionner un type. Si un champ est manquant une erreur est affichée. Une fois le bouton « crée » appuyé, la batterie se retrouve dans la liste. Si le bouton annuler est pressé le modal se ferme. | OK |
| Affichage de la liste des batteries | L’utilisateur peut afficher la liste de ses batteries en cliquant sur le bouton « mes batteries » dans le menu | URL : <http://localhost/batteries>  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01 | La batterie crée à l’étape précédente est visible dans la liste des batteries de l’utilisateur. | OK |
| Création d’une batterie lorsque l’utilisateur en possède au moins une | L’utilisateur est capable de créer une batterie si il en possède déjà une dans sa liste. | URL : <http://localhost/batteries>  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Capacité : 2600  Cellules : 4  C-Rate : 40  Type : LiPo  Appuyer sur le bouton « créer » pour accéder au modal de création. | Le modal s’ouvre, nous pouvons entrer les valeurs et créer la batterie en appuyant sur le bouton « créer ». La batterie est ensuite affichée dans la liste. Le bouton annuler ferme le modal. | OK |
| Suppression d’une batterie | L’utilisateur est capable de supprimer la batterie sélectionnée pour l’enlever de sa liste. | URL : <http://localhost/batteries>  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Batterie : LiPo 2600 4S 40C  Appuyer sur le menu  ‘’…’’ puis sur l’option supprimer. Confirmer ensuite l’action en appuyant sur « supprimer ». | La batterie est bien supprimée lorsque nous validons notre action depuis le modal. Un toast est affiché pour nous indiquer que la batterie a bien été supprimée. Si nous annulons notre action le modal se ferme et la batterie n’est pas supprimée. | OK |
| Modification d’une batterie | L’utilisateur est capable de modifier les valeurs de la batterie sélectionnée. | URL : <http://localhost/batteries>  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Batterie : LiPo 800 3S 25C  LiPo -> NiMH  Appuyer sur le menu  ‘’…’’ puis sur l’option modifier. Modifier les valeurs souhaitées puis appuyer sur « sauver ». | Le modal d’édition s’ouvre et nous avons la possibilité de modifier les valeurs. Une fois le bouton « sauver » cliqué, le modal se ferme et la batterie est mise à jour. Un toast nous indique que l’action à bien été réalisée. |  |
| Création d’un moteur lorsque l’utilisateur ne n’en possède pas encore | L’utilisateur est capable de créer un moteur si il ne n’en possède pas encore en cliquant sur un bouton. | URL : [http://localhost/](http://localhost/batteries)engines  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Nom : Fiala B2-FS  Type : thermique  Puissance : 85  Fréquence : 4t  Carburant : 95 SP 3.3%  Poids : 2940 image : ./tests/images/fiala-85.jpg  Appuyer sur le bouton « créer » pour accéder au formulaire de création. Remplir ensuite les champs puis envoyer. | Le formulaire s’ouvre et nous pouvons entrer les valeurs. En cliquant sur « créer » nous sommes redirigés sur la liste des moteurs et le nouveau moteur est affiché. | OK |
| Création d’un moteur lorsque l’utilisateur en possède au moins un | L’utilisateur est capable de créer un moteur si il en possède déjà un dans sa liste. | URL : [http://localhost/](http://localhost/batteries)engines  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Nom : T-Motor AM40  Type : électrique  Fréquence : 1500  Alimentation: 2-3 S  Poids : 22 image : ./tests/images/AM40.jpg  Appuyer sur le bouton « créer » pour accéder au formulaire de création. Remplir ensuite les champs puis envoyer. | Le formulaire s’ouvre et nous pouvons entrer les valeurs. En cliquant sur « créer » nous sommes redirigés sur la liste des moteurs et le nouveau moteur est affiché. | OK |
| Affichage de la liste des moteurs | L’utilisateur peut afficher la liste de ses moteurs en cliquant sur le bouton dans le menu | URL : [http://localhost/](http://localhost/batteries)engines  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01 | Les deux moteurs crées précédemment sont bien affiché dans la liste des moteurs. | OK |
| Suppression d’un moteur | L’utilisateur est capable de supprimer le moteur sélectionné pour l’enlever de sa liste. | URL : [http://localhost/](http://localhost/batteries)engines  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Nom du moteur : T-Motor AM40  Appuyer sur le menu  ‘’…’’ puis sur l’option supprimer. Confirmer ensuite l’action en appuyant sur « supprimer ». | Le moteur est bien supprimé lorsque nous validons notre action depuis le modal. Un toast est affiché pour nous indiquer que le moteur a bien été supprimé. Si nous annulons notre action le modal se ferme et le moteur n’est pas supprimé. | OK |
| Modification d’un moteur | L’utilisateur est capable de modifier les valeurs du moteur sélectionné. | URL : [http://localhost/](http://localhost/batteries)engines  Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Nom du moteur : Fiala B2-FS  Puissance : 120  Appuyer sur le menu  ‘’…’’ puis sur l’option modifier. Modifier les valeurs souhaitées puis appuyer sur « sauver ». | Le modal d’édition s’ouvre et nous avons la possibilité de modifier les valeurs. Une fois le bouton « sauver » cliqué, le modal se ferme et le moteur est mis à jour. Un toast nous indique que l’action à bien été réalisée. | OK |
| Accéder aux paramètres utilisateurs / gestion utilisateur | Il est possible d’accéder au paramètres utilisateurs/à la gestion de l’utilisateur | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Ouvrir le menu puis appuyer sur « mes paramètres » | La page de paramètres utilisateurs s’affiche | OK |
| Modification des données utilisateur avec des données correctes | Il est possible de modifier les données utilisateurs avec des données valides. | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Modifier le nom d’utilisateur en « DoeJ » | Le nom d’utilisateur est bien mis à jour. Un toast s’affiche pour nous confirmer la modification. | OK |
| Modification du nom d’utilisateur avec un nom d’utilisateur déjà existant | La modification du nom d’utilisateur n’est pas faite et une erreur est affichée à l’utilisateur lui indiquant que le nom d’utilisateur est déjà pris. | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Modifier le nom d’utilisateur en « R0kkxSynetique » | Le nom d’utilisateur n’est pas modifié et un toast s’affiche pour nous indiquer que le pseudo est déjà pris. | OK |
| Modification du mot de passe avec les bonnes informations | La modification du mot de passe en entrant le bon mot de passe actuel, le nouveau mot de passe et sa confirmation change effectivement le mot de passe | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Nouveau mot de passe : 01doe  Confirmation : 01doe | Le mot de passe est changé et nous pouvons nous reconnecter avec le nouveau. | OK |
| Modification du mot de passe avec le mauvais mot de passe actuel | Il est impossible pour l’utilisateur de modifier son mot de passe et un message d’erreur est affiché lui indiquant qu’il y’a un problème. | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : jdoe01  Nouveau mot de passe : johnD01  Confirmation : johnD01 | Le mot de passe n’est pas changé et un toast s’affiche pour nous informer que le mot de passe actuel n’est pas correct. | OK |
| Modification du mot de passe avec une confirmation incorrecte | Il est impossible pour l’utilisateur de modifier son mot de passe et un message d’erreur est affiché lui indiquant qu’il y’a un problème. | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : 01doe  Nouveau mot de passe : johnD01  Confirmation : john | Le mot de passe n’est pas changé et un toast s’affiche pour nous informer que les deux mot de passe ne sont pas identique. | OK |
| Modification du mot de passe avec des champs vides | Il est impossible pour l’utilisateur de modifier son mot de passe et un message d’erreur est affiché lui indiquant qu’il y’a un problème. | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : 01doe  Nouveau mot de passe : johnD01  Confirmation : - | Une erreur est affichée demandant de remplir le champ vide. | OK |
| Génération d’un QR code fonctionnel | Lorsque l’utilisateur clique sur le bouton « Imprimer QR », le QR code du modèle s’affiche dans la prévisualisation et redirige vers un lien de l’application | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : 01doe  Accéder à un modèle depuis « mes modèles » puis cliquer sur le menu ‘’…’’ puis sur « Imprimer QR ». | La prévisualisation d’impression s’ouvre et le QR code est affiché. Le QR code peut être lu par un smartphone et redirige vers un URL de aeromodel.mycpnv.ch . | OK |
| Impression d’un QR Code d’un modèle | Le QR code du modèle peut être imprimé depuis la prévisualisation | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : 01doe  Accéder à un modèle depuis « mes modèles » puis cliquer sur le menu ‘’…’’ puis sur « Imprimer QR ». Cliquer ensuite sur imprimer | Le QR code s’imprime sur l’imprimante sélectionnée. | OK |
| Accès à un modèle via son QR Code | Il est possible d’accéder à un modèle via son QR Code sans devoir s’authentifier | Utilisateur : Aucun  Se déconnecter de l’utilisateur actuel depuis le smartphone puis scanner le QR code. | La page invitée du modèle est affichée et nous n’avons aucunes actions possibles autre que la lecture. | OK |
| Suppression d’un compte utilisateur | L’utilisateur est capable de supprimer son compte depuis ses paramètres | Utilisateur: [john.doe@aero.ch](mailto:john.doe@aero.ch) Mot de passe : 01doe  Accéder aux paramètres, cliquer sur supprimer mon compte, confirmer la suppression. | Le compte de l’utilisateur est supprimé et se fait déconnecter de l’application. Il est impossible de se reconnecter avec les identifiants de connexion. | OK |

## Erreurs restantes

Il y’a deux chose auxquelles il faudrait faire attention.

Lors de l’édition d’un modèle ou d’un utilisateur, puisque la méthode PUT ne supporte pas correctement les requêtes ‘multipart’, il est impossible d’utiliser PUT pour mettre à jour une image en même temps qu’un formulaire. Il faudrait donc override la route de mise à jour du contrôleur de ressource « RcModelController » et la remplacer par une route POST pour corriger ce problème.

L’autre chose, il n’y a pas de vérification si l’utilisateur change l’id de l’URL d’un modèle par exemple. Cela implique que l’utilisateur qui exécute cette manipulation peut accéder aux données d’un autre et les modifier, supprimer, etc…

## Liste des documents fournis

# Conclusions

## Objectifs

### Atteints

Tous les objectifs demandés ont été complétés. C’est-à-dire que l’application permet à un utilisateur de créer, modifier, supprimer et lier un moteur ou une batterie. Elle permet aussi de modifier et supprimer un modèle. Les modèles possèdent des QR codes pouvant être généré et imprimé pour rediriger les utilisateurs invités sur la consultation du modèle. Et finalement, les utilisateurs peuvent changer de mot de passe ou même supprimer leur compte. Tout ça est disponible dans une interface responsive smartphone et pc.

### non-atteints

Aucuns objectifs n’ont pas pu être atteints.

## Points positifs

L’implémentation des nouvelles fonctionnalités à été très formatif. De plus, lors de ce projet, le planning mis à disposition m’a permis de travailler dans une bonne harmonie et m’a permis de correctement planifier le projet. Malgré les contre-temps rencontré, je n’ai jamais été en grande surcharge de travail et ai toujours trouvé des solutions.

## Points négatifs

Je regrette d’avoir travaillé sur des bases qui n’étaient pas bonnes car, il y’a encore maintenant, du code spaghetti ou non optimisé. En plus de cela le frontend a été lésé et ne possède approximativement aucune animation ce qui rends l’application un peu brutale lors de l’utilisation de celle-ci.

## Difficultés

Durant ce projet, j’ai rencontré de petits contre-temps avec les librairies de QR code, d’impression et de toasts. Lors de l’implémentation des QR codes et de l’impression de ceux-ci, je n’arrivais pas à en générer un et à l’afficher dans la prévisualisation d’impression. Ce souci provenait du fait que je ne voulais pas stocker mes QR codes dans le storage du server mais les générer en temps réel. Je n’arrivais donc pas à transmettre le svg généré dans ma prévisualisation. J’ai donc décider de ne pas générer le QR code à chaque fois mais vérifier si il existe déjà et si ce n’est pas le cas je le créer. De cette manière, j’ai pu donner à ma prévisualisation un fichier image au lieu d’un string svg.

L’autre contre-temps est arrivé lors de la recherche et de l’implémentation d’une librairie de toast qui soit compatible avec vue et qui me permette d’avoir une liberté de customisation. La librairie vue-toastification est très agréable à l’utilisation mais, à un petit défaut lors de l’utilisation de ceux-ci au chargement d’une page. Parfois il faut appeler la création de toast après le chargement de la page avec une requête d’animation pour déclencher la création du toast après avoir chargé l’entièreté de la page.

Un des autres contre-temps que j’ai eu était avec la mise à jour des modèles et du profil utilisateur car ils possèdent chacun d’eux une image. Avec les limitations de la méthode PUT et des requêtes ‘multipart’, il m’est impossible d’utiliser les routes crées avec le resource controller de Laravel.

Finalement, la dernière difficulté à été lors de la modification de mot de passe. Puisque j’utilise le middleware Sanctum, si le mot de passe de l’utilisateur change et ne correspond plus à celui de la session, l’utilisateur est automatiquement déconnecté. Je n’ai pas réussi à le reconnecter automatiquement après avoir changé son mot de passe.

## Suites

Pour que l’application puisse être agréable à utiliser il lui faudrait quelques ajouts.  
Par exemple, il serait agréable de pouvoir ajouter, modifier et supprimer ses propres hélices.

Suite au retour d’un utilisateur, un autre ajout serait la répertoriassions des récepteurs des modèles réduits. Il m’a aussi fait parvenir un souhait d’implémenter un code couleur des pages pour se repérer dans l’application. Toujours en lien avec les couleurs, ajouter un code couleur pour repérer les modèles possédant un moteur thermique, électrique ou sans moteurs. Ces couleurs devraient être implémentés dans la consultation des moteurs et des batteries. Dernièrement, l’ajout de présélection sur le type de carburant ferait gagner du temps.

Toutes ces modifications rendraient l’application plus pertinente.

Il y’a énormément d’autres possibilités d’ajout comme le suivi des batteries, le répertoriage des réparations, des crashs, des vols, etc…

# Glossaire

# Annexes

## Résumé du rapport du TPI

Le projet a été débuté le 2 mai 2023 et s’est terminé le 2 juin 2023. Il est dans la continuité de mon projet de Pré-TPI et apporte de nouvelles fonctionnalités à celui-ci. Ce projet avait donc pour but d’ajouter une gestion accrue des modèles réduits, des moteurs et des batteries. En plus de cela, il a fallu ajouter une gestion de compte utilisateur et implémenter des QR codes pour les modèles. Evidemment, le tout devait être responsive et devait tenir dans un horaire de 90 heures de travail.

Tout a commencé avec l’analyse et la planification qui s’est faite durant les 3 premier jours. Durant ce temps, les tâches ont été préparée et placées dans des sprints. Une fois les tâches estimée et planifiée, les wireframes puis les maquettes ont été mis à jour pour subvenir aux nouveaux besoins du projet. Dès lors que les maquettes étaient finies, le MCD puis le MLD ont eux aussi été mis à jour. À la suite de cela les migrations Laravel ont été créée pour modifier la base de données. C’est seulement à ce moment où le backend a commencé à être modifié. Tout a commencé avec les fonctionnalités relativement simple et rapide à implémenter. La gestion des modèles a été la première étape suivie, la gestion des batteries et finalement celle des moteurs. A la fin de l’implémentation de ces fonctionnalités, j’ai eu un jour d’avance sur mon planning. A partir d’ici j’ai commencé l’implémentation des fonctionnalités plus complexe. La génération et l’impression des QR codes m’a pris quelques heures de plus que prévu mais je suis resté dans mon planning. Après cela, l’implémentation de l’authentification via un token s’est fait assez rapidement une fois la solution de génération trouvée. Finalement j’ai terminé les fonctionnalités principales avec la gestion de compte utilisateur. Ensuite, j’ai fait énormément de frontend pour corriger le responsive des pages de l’application. Et, une fois le tout fini j’ai fait des tests complets du site pour trouver tous les potentiels bugs. Finalement la dernière semaine m’a permis de compléter et mettre en page ma documentation.

### Résultats

## Sources – Bibliographie

## Journal de travail

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet