

Projet de diplôme technicien 2020

Tom Ryser | Technicien ES info. | Lundi, 8 juin 2020

Table des matières

[Table des figures 3](#_Toc39757945)

[Résumé 4](#_Toc39757946)

[Abstract 4](#_Toc39757947)

[Contexte 4](#_Toc39757948)

[Environnement 5](#_Toc39757949)

[Laragon 5](#_Toc39757950)

[Laravel 5](#_Toc39757951)

[Axure RP 5](#_Toc39757952)

[Postman 5](#_Toc39757953)

[Visual studio code 5](#_Toc39757954)

[Cahier des charges 5](#_Toc39757955)

[Analyse de l’existant 6](#_Toc39757956)

[divelogs 6](#_Toc39757957)

[deepblu 7](#_Toc39757958)

[logbook.diviac 7](#_Toc39757959)

[Analyse fonctionnelle 8](#_Toc39757960)

[Analyse organique 8](#_Toc39757961)

[Mise en place 8](#_Toc39757962)

[Github 8](#_Toc39757963)

[Trello 8](#_Toc39757964)

[Maquettes 8](#_Toc39757965)

[Pré-travail 9](#_Toc39757966)

[Recherche 9](#_Toc39757967)

[En-tête de fichier 9](#_Toc39757968)

[Diagrammes d’activités 9](#_Toc39757969)

[Base de données 10](#_Toc39757970)

[UML 10](#_Toc39757971)

[Privilèges 10](#_Toc39757972)

[API 11](#_Toc39757973)

[Importation d’une plongée 11](#_Toc39757974)

[Récupération d’une ou plusieurs plongées 11](#_Toc39757975)

[Conclusion 13](#_Toc39757976)

[Glossaire 15](#_Toc39757977)

[Annexes 15](#_Toc39757978)

[Installation 16](#_Toc39757979)

[Tests 18](#_Toc39757980)

[Api 18](#_Toc39757981)

# Table des figures

[Figure 1 Interface Divelogs.de 6](#_Toc39760607)

[Figure 2 Example de modification 6](https://d.docs.live.net/e0970d4b4942baac/Documents/documentationTechnique_diplome_2020.docx#_Toc39760608)

[Figure 3 Graph de plongée manuscrite 7](https://d.docs.live.net/e0970d4b4942baac/Documents/documentationTechnique_diplome_2020.docx#_Toc39760609)

[Figure 4 Graph de plongée avec data 7](https://d.docs.live.net/e0970d4b4942baac/Documents/documentationTechnique_diplome_2020.docx#_Toc39760610)

[Figure 5 Abonnement logbook.diviac 7](https://d.docs.live.net/e0970d4b4942baac/Documents/documentationTechnique_diplome_2020.docx#_Toc39760611)

[Figure 6 Aperçu model créer avec Workbench 10](#_Toc39760612)

[Figure 7 Json DiveController@index 12](#_Toc39760613)

[Figure 8 install heidi.sql 16](#_Toc39760614)

[Figure 9 install execute. sql 17](#_Toc39760615)

# Résumé

Dans le monde de la plongée subaquatique il y a de nombreuses personnes, même s’ils ne se connaissent pas, ils partagent tous une passion pour les fonds marins. Le but de ce projet est de permettre aux plongeurs de partagé leurs expériences ainsi que leurs découvertes. À travers cette application, ils pourront enregistrer leurs plongées et les partagées au reste de la communauté. Il sera aussi possible de partager des sites de plongées ainsi que des clubs.

La difficulté de ce projet réside dans l’intégration des données de plongées récupérée sur les différents ordinateurs de plongées ainsi que leur conversion dans le format UDDF[[1]](#footnote-1).

Ce document comporte l’ensemble des aspects techniques liés à ce projet, ainsi que certaines procédures pour le maintenir.

# Abstract

In the world of underwater diving there are many people, even if they don't know each other, they all share a passion for the seabed. The aim of this project is to allow divers to share their experiences and discoveries. Through this application, they will be able to record their dives and share them with the rest of the community. It will also be possible to share dive sites and clubs.

The difficulty of this project lies in the integration of the dive data retrieved from the different dive computers and their conversion into the ... format.

This document contains all the technical aspects related to this project, as well as some procedures to maintain it.

# Contexte

Ces dernières années la plongée subaquatique est devenue plus accessible, et les équipements ne cessent de se moderniser et aujourd’hui nous sommes à la mode du sans fil et du partage, ce qui rend le besoin d’application pour gérer ces données plus importantes que jamais

# Environnement

## Laragon

Laragon est un serveur local pour le développement, qui permet la création et la gestion de projet laravel simplement.

J’ai non seulement choisi Laragon pour Laravel, mais aussi car contrairement à EasyPhp et Wamp je n’ai jamais eu de bug.

## Laravel

Laravel est un framework web open-source écrit en PHP. Laravel tente d’alléger le développement en facilitant les tâches courantes utilisées dans la majorité des projets Web, comme l’authentification, le routage, les sessions et la mise en cache.

Laravel intègre des concepts modernes de développement tels que l’inversion de contrôle, un système de migration de base de données et un support de test unitaire intégré.

## Axure RP

Axure RP, est un outil logiciel de wireframing, de prototypage rapide, de documentation et de spécification destiné aux applications web, mobiles et de bureau.

## Postman

Postman est une plateforme pour le développement d’API, elle permet de designer, construire et tester des APIs.

## Visual studio code

Visual studio code est un IDE que j’utilise pour tous mes projets. Il est open source ce qui permet d’avoir une grande quantité d’add-on.

Les add-on utilisé dans le cadre du projet :

* Laravel Snippets
* Markdown All in One
* Material Icon Theme
* PHP Debug

Visual studio code intègre aussi des fonctionnalités pour GitHub

# Cahier des charges

# Analyse de l’existant

## divelogs

Divelogs est un carnet de plongée en ligne offrant une interface très chargée

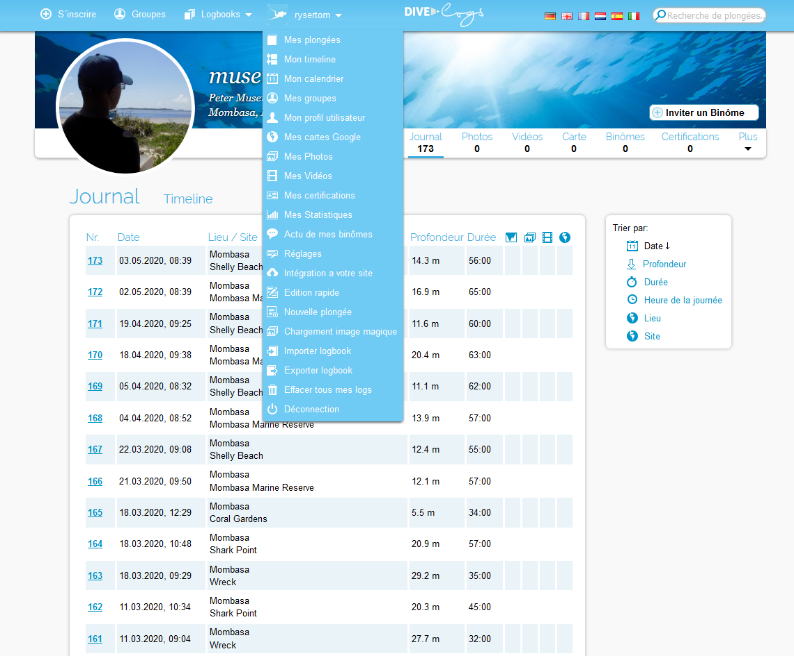


Figure 1 Interface Divelogs.de

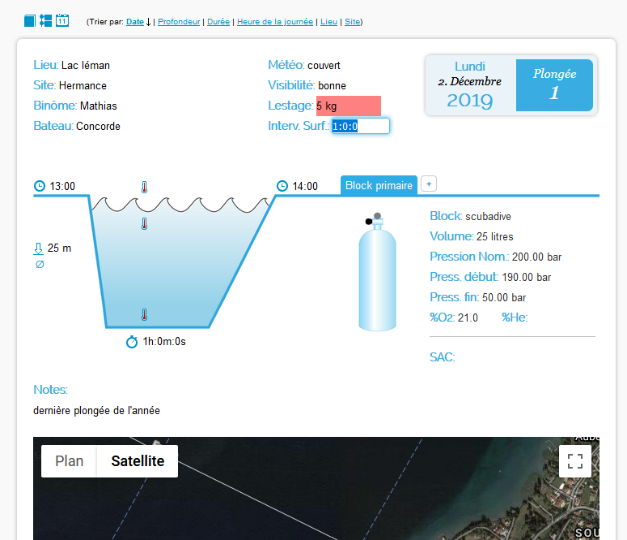
Lors de mes tentatives d’import de logs XML provenant de l’application Subsurface, je me retrouve à chaque fois bloqué sur une page blanche. Mais Subsurface offre la possibilité d’exporter les informations directement sur le serveur de Divelogs.

Figure 2 Example de modification

L’édition d’une plongée est simple, il suffit de cliquer sur l’élément que l’on souhaite modifier, et la modification est enregistrée automatiquement à la fin de son édition, mais aucun moyen de savoir l’état de celle-ci (échec ou réussite)

## deepblu

Deepblu est une application web orienté réseau social permettant de renseigner ces plongées ainsi que de partager tout type d’informations.

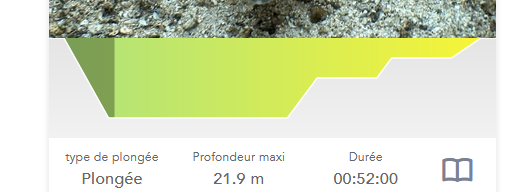
Il existe aussi une application mobile (IOS et Android)

Figure 3 Graph de plongée manuscrite

Il propose aussi l’importation de logs provenant d’ordinateurs de plongées, mais ils ne supportent que les formats de Subsurface. Lors de l’importation d’un log UDDF, l’utilisateur voit apparaître un message lui disant que le traitement peut durer jusqu’ à 60 minutes. Après l’importation l’on reçoit une notification sur le site, et l’on passe à la modification de la plongée importée avant de pouvoir finaliser l’enregistrement.

Sur la page d’une plongée l’on peut voir les informations de la plongée ainsi qu’un graphique de la plongée (profondeur x temps) et ce que je présume être une carte avec la position de la plongée.

Figure 4 Graph de plongée avec data

## logbook.diviac

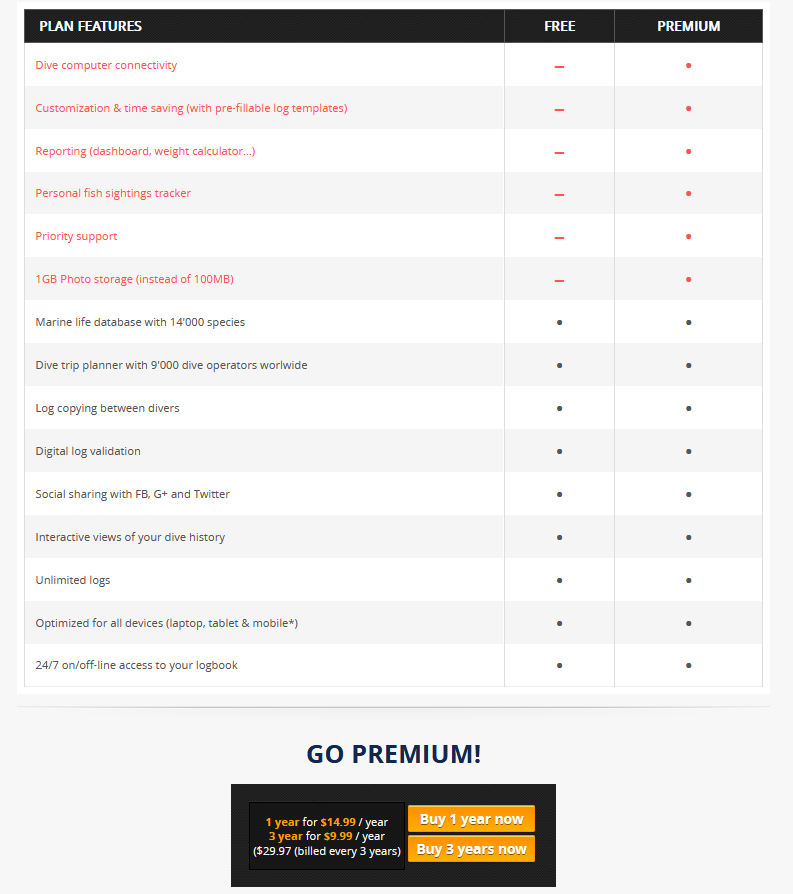
Diviac est un carnet de plongée en ligne qui offre un system pour les instructeurs, qui leurs permet de copier des plongées dans les carnets de leurs élèves.

Figure 5 Abonnement logbook.diviac

Diviac est aussi la seule application à proposer un service premium.

L’importation de fichier n’est malheureusement pas disponible pour les utilisateurs gratuits.

# Analyse fonctionnelle

# Analyse organique

## Mise en place

### Github

Afin d’éviter les problèmes de pertes et faciliter le partage j’ai placer mon projet sur GitHub

### Trello

Trello me permet de savoir les taches qu’il me reste, ainsi que celle qui sont déjà finie.

Il est composé de 3 colonnes :

* A faire
* En cours
* Fait

### Maquettes

## Pré-travail

### Recherche

Afin de préparer le projet j’ai fait des recherches sur les différents types d’exports de fichiers, provenant des logiciels fournis par les constructeurs d’ordinateurs de plongées.

A ce jour la plupart des constructeurs s’orientent vers le Bluetooth pour exporter leurs données. Mais il n’y a pas encore de format standard.

J’ai donc décidé d’utiliser le format UDDF comme format pivot, car il est le format le plus complet et maintenu dans le domaine.

J’ai par la suite sélectionné parmi les formats que j’ai pu récupérer, les 3 qui me semblait les plus pertinent à convertir.

### En-tête de fichier

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUTEUR      : Tom Ryser

LIEU        : CFPT Informatique Genève

DATE        : Avril 2020

TITRE PROJET: ScubaRecords

VERSION     : 1.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

C’est en-tête me permet donc de facilement repérer les fichiers que j’ai créé et développer, ainsi que voir leur version dans un futur ou il y aura des mises à jour de l’application.

## Diagrammes d’activités

# Base de données

## UML

Pour créer la base de données plus facilement j’ai décidé d’utiliser Workbench pour créer un model de la base de données.

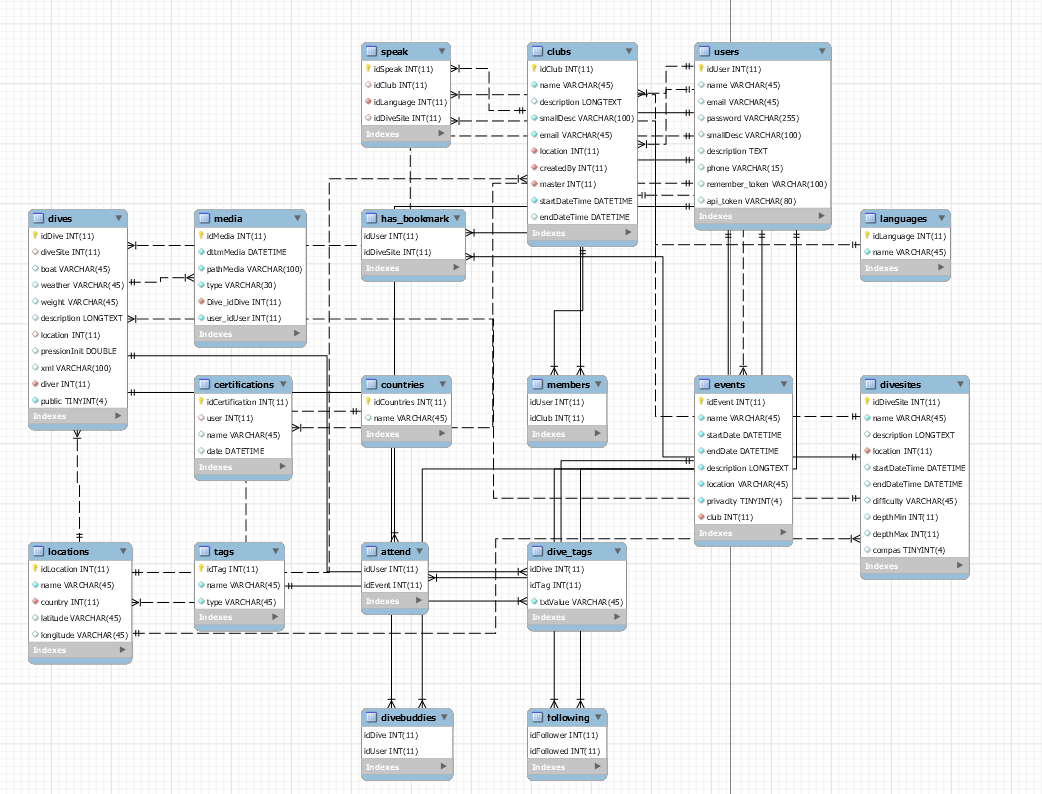


Figure 6 Aperçu model créer avec Workbench

## Privilèges

Pour accéder à la base de données, il faut utiliser les privilèges suivants :

* Username : **ScubaRecords**
* Password : **o65uCeEngfo(ttzLmm2pFBij**

# API

L’api est divisée en plusieurs parties :

1. Les Routes
   1. \routes\api.php
   2. \routes\web.php
2. Les Controllers
   1. \app\http\Controllers\nom\_du\_fichier.php
3. Les models
   1. \app\nom\_du\_fichier.php

## Importation d’une plongée

## Récupération d’une ou plusieurs plongées

### Eloquent

public function index(){

        $dives = DB::table('dive\_tags')

        ->selectRaw('

            DISTINCT dives.\*, GROUP\_CONCAT(

            DISTINCT CONCAT(tags.name,":",dive\_tags.txtValue)

            ORDER BY dive\_tags.idTag

            SEPARATOR ";") AS tags,

            users.name as username,

            locations.name as locationName,

            diveSites.name as diveSiteName

            ')

        ->join('tags', 'tags.idTag', '=', 'dive\_tags.idTag')

        ->join('dives', 'dives.idDive', '=', 'dive\_tags.idDive')

        ->join('users', 'users.idUser', '=', 'dives.diver')

        ->join('locations', 'locations.idLocation', '=', 'dives.location')

        ->join('diveSites', 'diveSites.idDiveSite', '=', 'dives.diveSite')

        ->where('dives.public', 1)

        ->orderBy('dives.idDive', 'DESC')

        ->groupBy('dives.idDive')

        ->get();

        $data = $this->serializeDive($dives->toArray(), 0);

        return view('dives')->with( 'data', json\_decode($data, true));

    }

Voici la requête Eloquent pour récupérer toutes les plongées publiques.

#### GROUP\_CONCAT

Est une fonction qui retourne sous forme de string concaténée les informations d’une table,

Ce qui me permet de récupérer toutes les informations des tags d’une plongée.

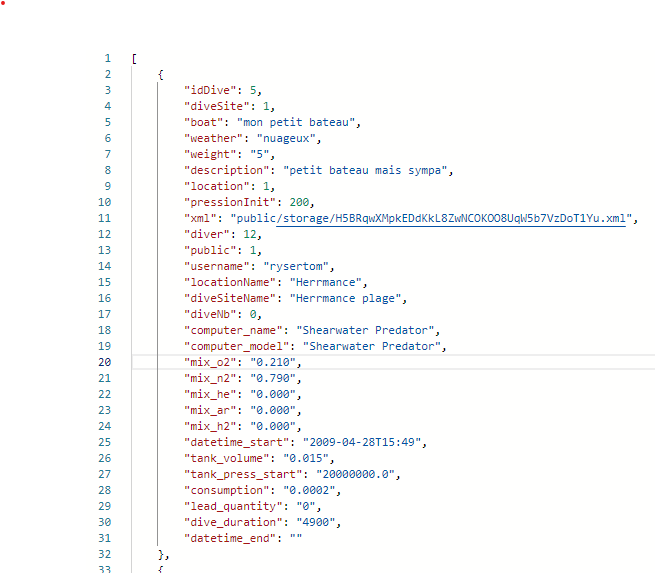


Figure 7 Json DiveController@index

# Raisonnement

## Journal de bord

Pour le suivi de mon projet j’ai décidé de faire un journal de bord en markdown, pour qu’il soit lisible depuis le GitHub.

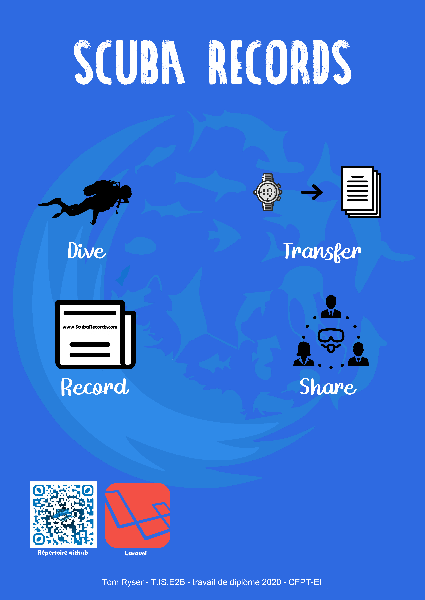
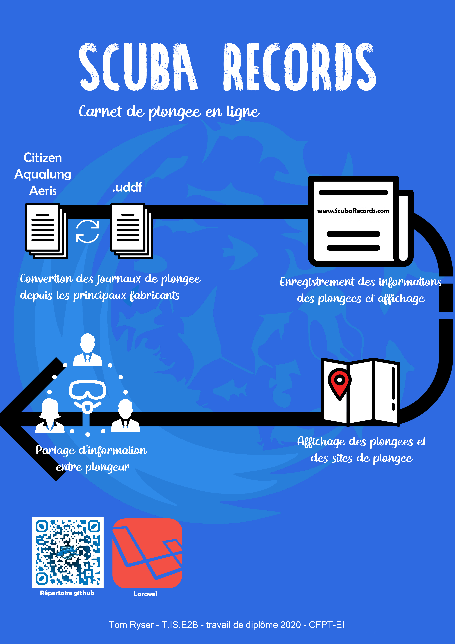
### Structure

### 

## Poster

V1, cette version était ambigüe, les gens pensent que je fais une application de transfère entre une montre et un site web

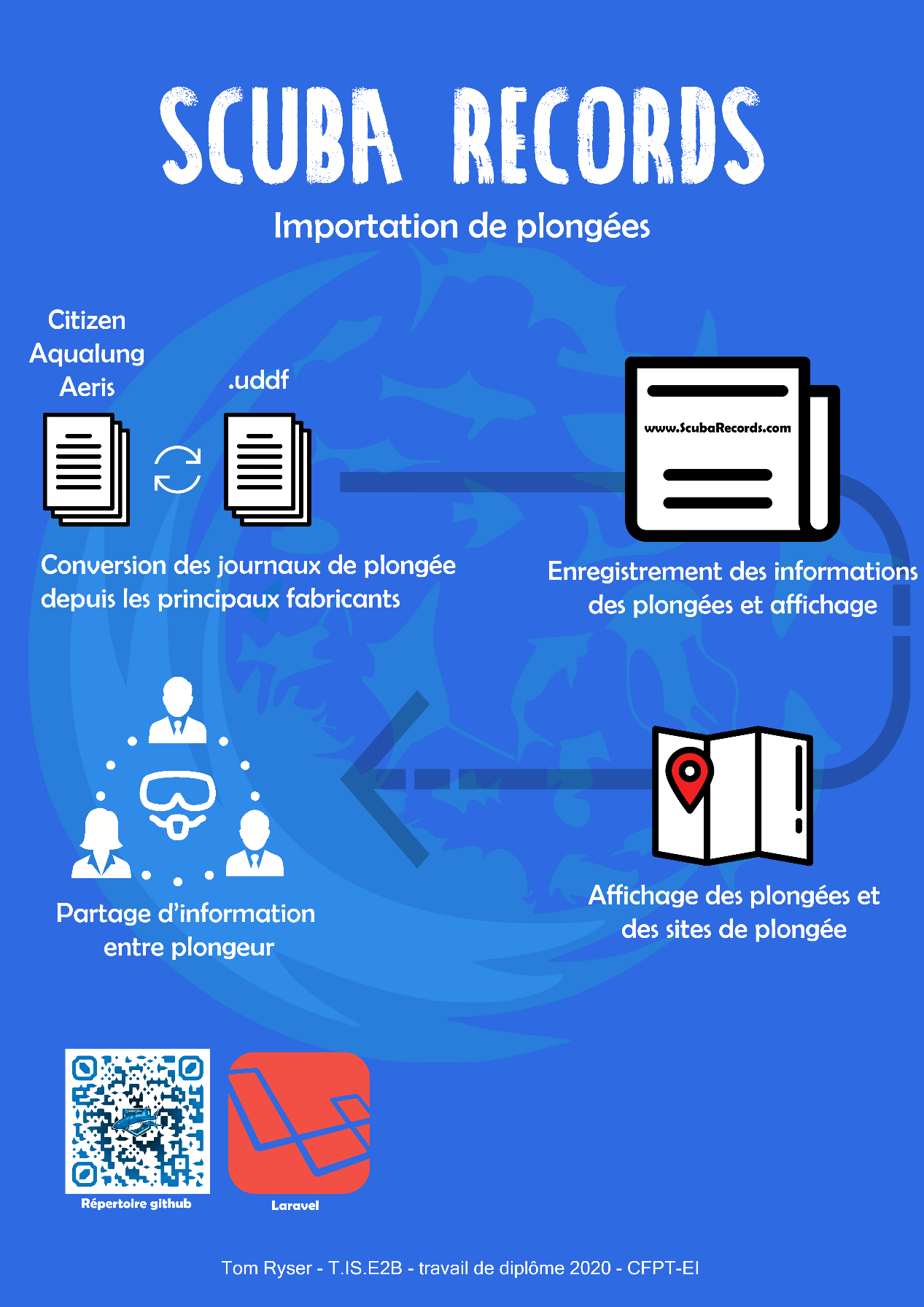
V2, est mieux compréhensible mais elle manque de direction.



V3, J’ai fait 2 essaie que j’ai montré à ma classe.

### Final

Pour finir j’ai fait encore quelques modifications mineures après retour de M. Mathieu



# Conclusion

# Glossaire

# Annexes

# Installation

1. Installer laragon
   1. <https://laragon.org/docs/install.html>
2. Cloner le répertoire github
   1. <https://github.com/Ryser-tom/ScubaRecords>
3. Placer le projet dans le répertoire C:\laragon\www
4. Lancer laragon
   1. Et cliquez sur « Start all »
   2. Puis clique droit dans laragon, sous MySQL et HeidiSQL

Une image contenant capture d’écran, ordinateur, moniteur, portable

Description générée automatiquement

Figure 8 install heidi.sql

* 1. Dans Heidi, cliquez sur Ouvrir
  2. Sous Fichier cliquez sur « Exécuter un fichier SQL… » et ouvrez le fichier « bddCreate.sql » se trouvant dans le dossier /Documentation/bdd du projet

Une image contenant capture d’écran

Description générée automatiquement

Figure 9 install execute. sql

* 1. Retournez sur laragon et cliquez sur « Terminal », et entrez les commandes suivantes :
     1. cd ScubaRecords-master
     2. composer install
     3. copy .env.example .env
     4. php artisan key:generate
  2. Pour finir faites un click droit dans laragon puis allez sous « www » et « ScubaRecords-master » pour être redirigé sur le site.

# Tests

## Api

Pour les tests de l’api j’utilise postman, la collection se trouve dans : /Documentation/ScubaRecords.postman\_collection.json

1. <https://subsurface.hohndel.narkive.com/p3HxfvcD/uddf-export-format> [↑](#footnote-ref-1)