

|  |
| --- |
| Gestion de cave |

Gfeller, Jérémy

SI-C4a

TPI fin 2018

Table des matières

[1 Analyse préliminaire 3](#_Toc516065591)

[*1.1* Introduction 3](#_Toc516065592)

[*1.2* Objectifs 3](#_Toc516065593)

[*1.3* Planification initiale 4](#_Toc516065594)

[2 Analyse / Conception 4](#_Toc516065595)

[*2.1* Concept 4](#_Toc516065596)

[2.1.1 Vue d’ensemble 4](#_Toc516065597)

[2.1.2 MCD 6](#_Toc516065598)

[*2.2* Stratégie de test 6](#_Toc516065599)

[*2.3* Risques techniques 7](#_Toc516065600)

[*2.4* Planification 8](#_Toc516065601)

[*2.5* Dossier de conception 9](#_Toc516065602)

[2.5.1 Maquettes 10](#_Toc516065603)

[2.5.2 Use cases / Scénarios 15](#_Toc516065604)

[2.5.3 MLD 21](#_Toc516065605)

[3 Réalisation 23](#_Toc516065606)

[3.1 Site web 23](#_Toc516065607)

[3.1.1 Nouvelle entrée en cave 23](#_Toc516065608)

[3.2 Entrée, sortie de bouteilles 24](#_Toc516065609)

[3.3 Stock 25](#_Toc516065610)

[3.4 Application mobile 26](#_Toc516065611)

[3.4.1 Entrée / sortie de bouteille 26](#_Toc516065612)

[3.4.2 Gérer le stock 30](#_Toc516065613)

[*3.5* Dossier de réalisation 32](#_Toc516065614)

[3.5.1 Partie site 32](#_Toc516065615)

[3.5.2 Partie application mobile 33](#_Toc516065616)

[3.6 Description des tests effectués 33](#_Toc516065617)

[3.7 Erreurs restantes 34](#_Toc516065618)

[3.8 Liste des documents fournis 34](#_Toc516065619)

[4 Conclusions 35](#_Toc516065620)

[4.1 Conclusion personnel 35](#_Toc516065621)

[5 Annexes 36](#_Toc516065622)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 36](#_Toc516065623)

[5.1.1 Situation de départ 36](#_Toc516065624)

[5.1.2 Mise en œuvre 36](#_Toc516065625)

[5.1.3 Résultats 36](#_Toc516065626)

[5.2 Glossaire 37](#_Toc516065627)

[*5.3* Sources – Bibliographie 37](#_Toc516065628)

[*5.4* Journal de bord 37](#_Toc516065629)

[*5.5* Manuel d'Installation 38](#_Toc516065630)

[5.5.1 Windows 38](#_Toc516065631)

[5.5.2 Mac 38](#_Toc516065632)

[*5.6* Aide externe 38](#_Toc516065633)

# Analyse préliminaire

## Introduction

Ce projet est réalisé dans le cadre du TPI final en dernière année au CPNV. Un client est venu demandé à l’école s’il était possible que quelqu’un fasse une application mobile et une partie d’un site web afin qu’il puisse gérer son stock de vin de manière simple et efficace. M. Louis Pache gère la cave à vins du cercle d’Yverdon-les-Bains.

Le cercle est un groupe de personnes ayant des centres d’intérêt commun, notamment celui du vin. Ils possèdent une cave à vins avec environ plusieurs centaines de bouteilles de vin.

Le but de ce projet est d’offrir une application mobile de gestion très simple du contenu de la cave. Il y aura aussi une partie web sur leur site internet (<http://www.cercledyverdon.ch>) qui permettra aussi la gestion des vins de manière assez simple. Leur site web est hébergé chez Infomaniak.

Ces nouvelles pages internet ne seront accessibles que par certains membres choisis par le webmaster.

## Objectifs

Voici la liste des objectifs qui m’ont été transmis pour ce projet :

Le caviste peut utiliser le site pour :

* Introduire un nouveau vin dans le système (base de données)
* Ajouter ou retirer des bouteilles d’un vin existant
* Imprimer les QR Codes des vins pour les coller sur les étagères (et non sur les bouteilles)
* Consulter le stock actuel
* Consulter le stock à une date donnée

Le caviste peut utiliser l’application mobile pour :

* Comparer le nombre théorique (dans la base de données du site) et le nombre effectif de bouteille (en cave) : il scanne le QR code dans la cave, l’application affiche le nombre de bouteilles théoriquement en stock
* Ajuster le stock : dans le cas où il constate une différence entre théorie et réalité, il ajuste le nombre théorique avec l’application
* Ajouter et retirer des bouteilles de vin existant

L’application mobile doit pouvoir fonctionner hors connexion :

* La cave étant un endroit difficilement atteignable par les réseaux WIFI ou 4G, les opérations de consultation et de mise à jour du stock doivent être possible hors connexion. Ce n’est qu’ensuite, lorsque la connexion est rétablie, que le caviste pourra demander la synchronisation du stock.

## Planification initiale

Le projet a commencé le 08 mai 2018 à 08h55. La fin du TPI est prévue pour le 7 juin à 10h35. Tous les livrables doivent être transmis avant cette date et heure finale.

# Analyse / Conception

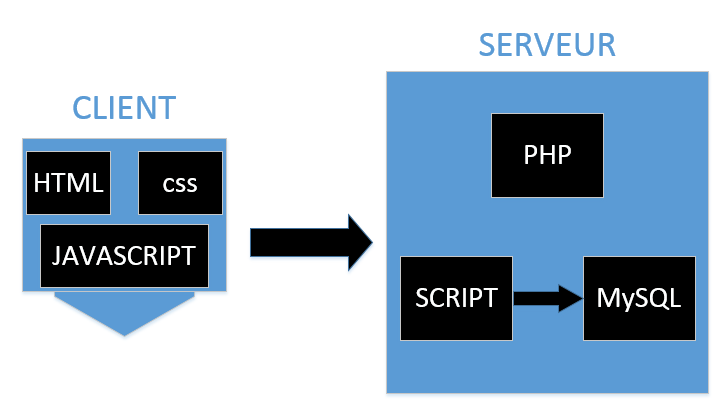
## Concept

### Vue d’ensemble

#### Vue d’ensemble du site web

Sur l’ordinateur du client, il y aura le site avec les différentes pages créées, qui s’affichera. Chaque action faite sera ensuite envoyée par une requête à la base de données afin de faire les modifications que le caviste voudra.

Pour la base de données, il y a un script qui la crée et qui ajoute quelques données à l’intérieur de celle-ci.



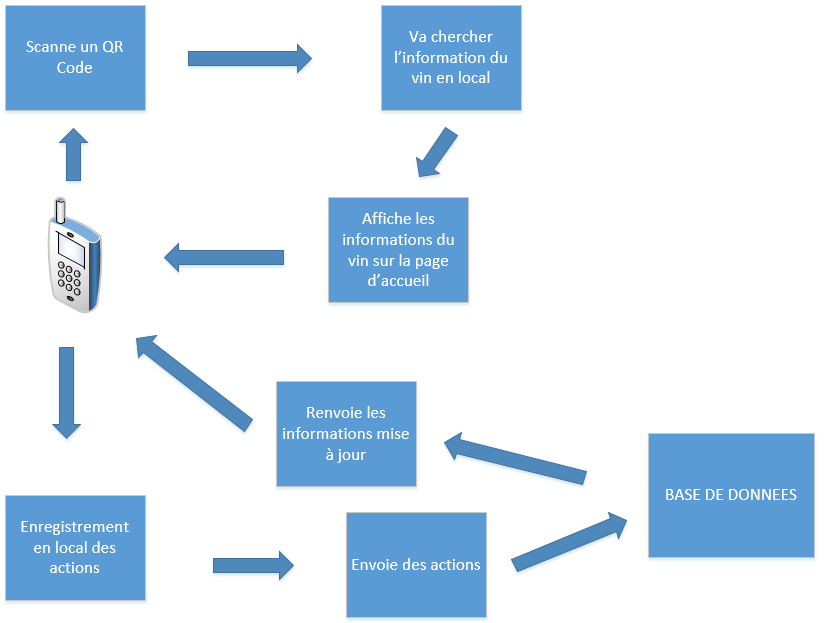
#### Vue d’ensemble de l’application mobile

Sur l’application mobile du caviste, les vins seront enregistrés en local. Ça lui permettra d’utiliser l’application lorsque la connexion sera absente.

Lorsque l’utilisateur synchronise ses actions, comme par exemple un mouvement ou un nouvel inventaire, avec la base de données, cette dernière lui renverra les vins et leurs quantités à jour.

En ce qui concerne les synchronisations, elles seront disponibles uniquement lorsqu’une connexion internet est établie.

Lors d’un scanne d’un QR Code, c’est la même chose que pour les actions, sauf qu’il va chercher les vins qu’il a en local pour faire afficher les informations de celui-ci.



### MCD

Voici le MCD de ma base de données

Suite à ce MCD ma base de données comprendra cinq tables en tout. Il y aura une table de plus qui viendra entre la table utilisateurs et la table millésime. Ce sera la table mouvement.

## Stratégie de test

Pour l’application mobile, les tests se feront en même temps que le développement, ce qui me permettra de combler un maximum de bug. M. Chavey a fait quelques tests de l’application et a trouvé des bugs, qu’on a corrigé en mettant en place des fonctions.

Les tests ont été effectués une partie sur un navigateur internet, sur un émulateur de smartphone et pour finir sur mon iPhone personnel.

Pour la partie site web, j’ai mis à disposition des champs pour les dates qui sont faciles à tester. Il faut que le champ ait une valeur à l’intérieur pour que l’action soit effectuée.

La plupart des champs présents sur le site ont comme indication le fait qu’il y ait au moins une valeur à l’intérieur.

## Risques techniques

Les risques techniques étaient principalement présents dans la partie mobile, une partie des fonctions avaient été vues durant le pré-TPI mais d’autres non, comme par exemple le fait que l’application doit être capable de fonctionner en mode hors connexion.

Mon plus gros problème sur la partie mobile a été d’implémenter la fonction permettant à l’application de fonctionner même en étant en mode hors connexion.

J’ai essayé une première fonction du Framework ionic qui s’appelle « SQLite » qui permet de créer une base de données en local sur le téléphone. Au bout d’un certain temps, avec M. Carrel, nous avons décidé de basculer sur une fonction qui s’appelle « storage » et de remplacer celle-ci testée précédemment.

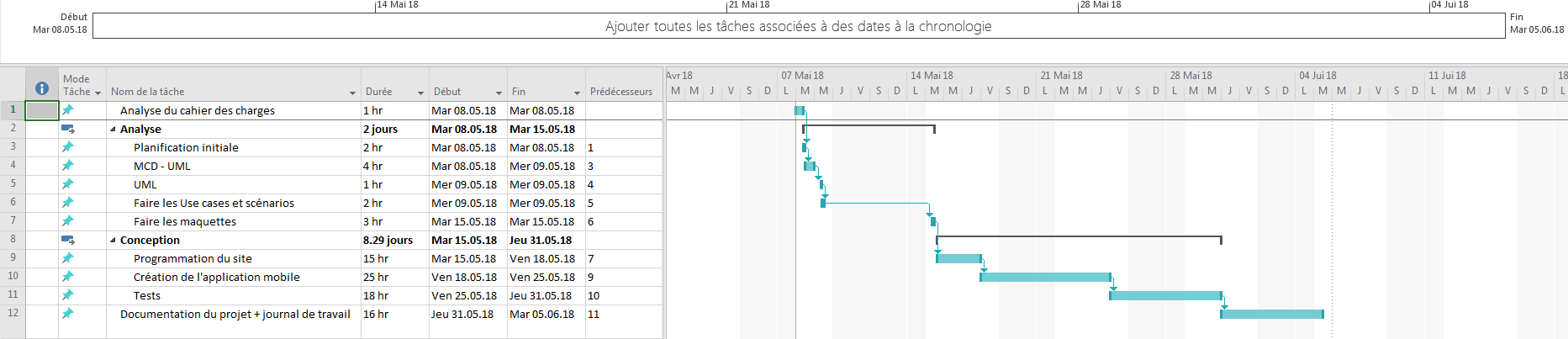
Cette nouvelle fonction permet de sauvegarder des données en local sur le téléphone.

Vu que nous avons changé en fin de projet de solution, j’ai fini par implémenter la dernière fonction choisie par M. Carrel et moi-même.

Ensuite à cause de cette fonction nous avons changé ma base de données une semaine avant la fin du TPI. Du coup j’ai dû refaire certaines fonctions du site et de l’application avec les modifications apportées à la base de données.

## Planification

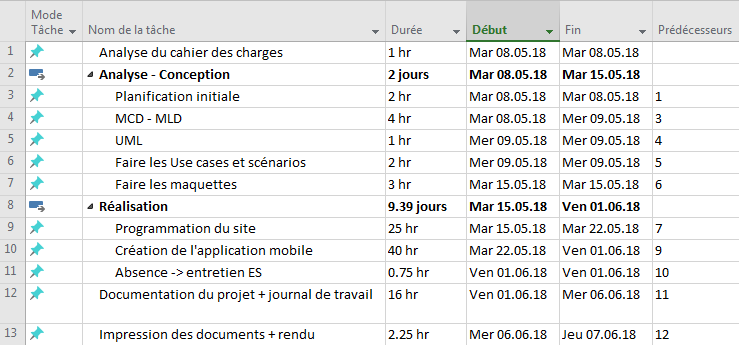
Planification initiale du projet réalisé sur Project :



Le temps d’une tâche sera mis à jour si elle prend plus, ou moins de temps, de ce qui est prévu ci-dessus.

La documentation du projet est tenue à jour hebdomadairement tandis que le journal de travail lui est tenu à jour journalièrement. Ils ne seront pas remplis le dernier jour du TPI comme indiqué dans le point 12 de la planification initiale ci-dessus.

Pendant tout le projet, la documentation est à donner une fois par semaine aux experts contrairement au journal de travail qui est à envoyer deux fois par semaine.

Planification mise à jour avec les valeurs modifiées

En fin de projet, la réalisation de l’application mobile a pris plus de temps que prévu, nous avons fait des modifications dans la base de données une semaine avant la fin du rendu du TPI. J’ai dû alors refaire des fonctions pour que l’application fonctionne correctement avec le changement fait. J’ai également dû faire des modifications sur la partie du site web.

## Dossier de conception

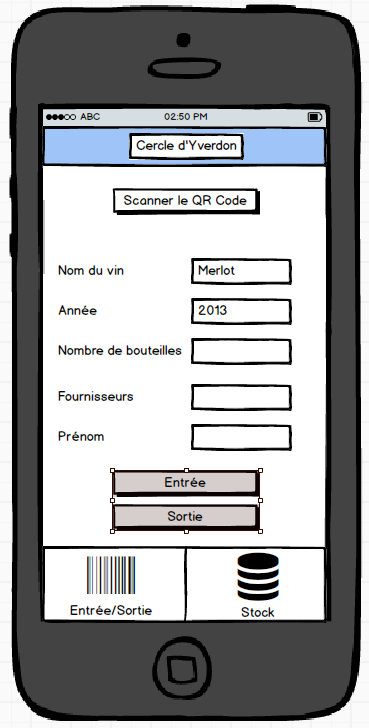
Pour le dossier de conception voici la liste des logiciels utilisés pendant ce TPI :

* Cmder, c’est une version améliorée du cmd qui est de base sur les machines Windows.
* Word, pour rédiger des documents comme la documentation du projet ou encore les use cases et scénarios.
* Excel, pour faire mon journal de travail.
* Visual Studio Code, qui me permet de développer les pages du site et l’application mobile.
* GitHub desktop, cette application permet de synchroniser mes fichiers modifiés sur mon répertoire GitHub. GitHub est un service web d'hébergement et de gestion de développement de logiciels.
* MySQL Workbench, il me permet de gérer ma base de données.
* Firefox, navigateur par défaut pour aller sur internet.
* Explorateur de fichier de Windows, me permettant d’aller chercher, de déplacer ou de télécharger des fichiers.
* Wamp, un serveur local pour pouvoir utiliser les fichiers au format PHP.
* Project, pouvoir créer la planification initiale du projet.
* Outil capture, permet de faire des captures d’écrans.
* Android Studio, elle permet de faire une émulation d’appareils Android, comme des tablettes ou des smartphones.

### Maquettes

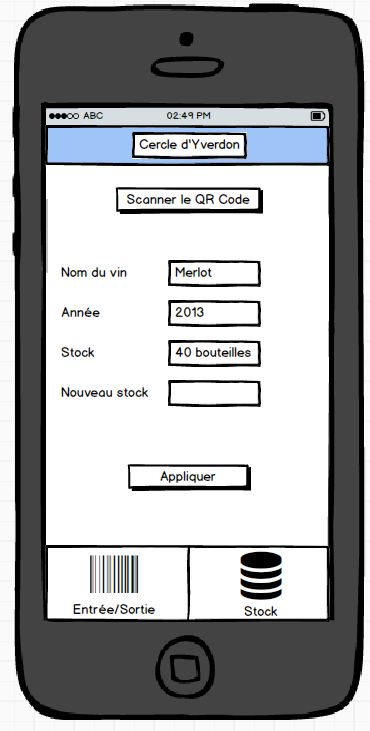
#### Entrée et sortie du vin

Cette page permet au caviste d’ajouter les bouteilles qui rentrent dans la cave et mettre à jour dans la base de données. Il pourra aussi faire l’inverse, enlever le nombre de bouteilles qu’il enlève de la cave et mettre à jour dans la base de données.



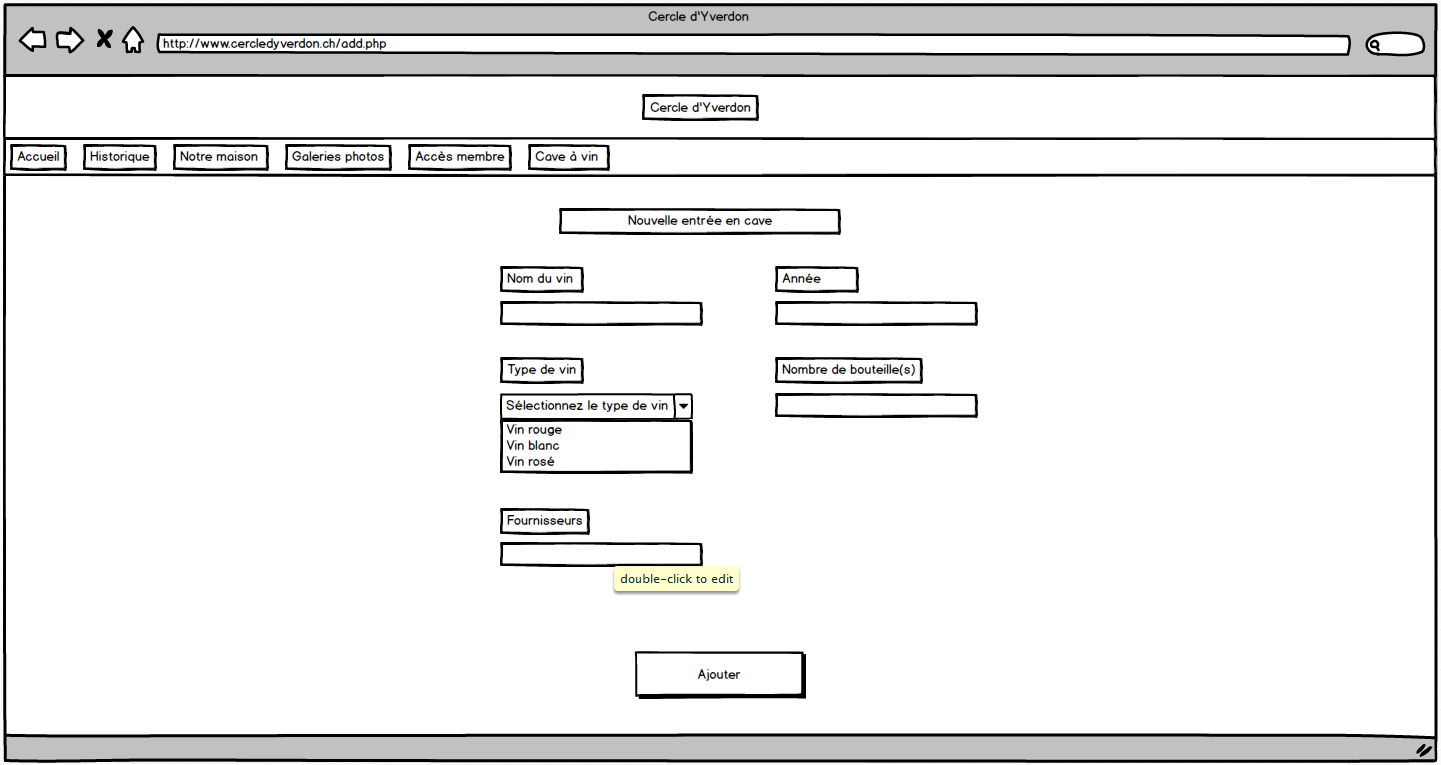
#### Gérer les stocks

Cette page permet au caviste de comparer le nombre de bouteilles présente en cave à celui qui est sur la base de données. S’il remarque une différence il y a possibilité de mettre à jour le stock dans la base de données.



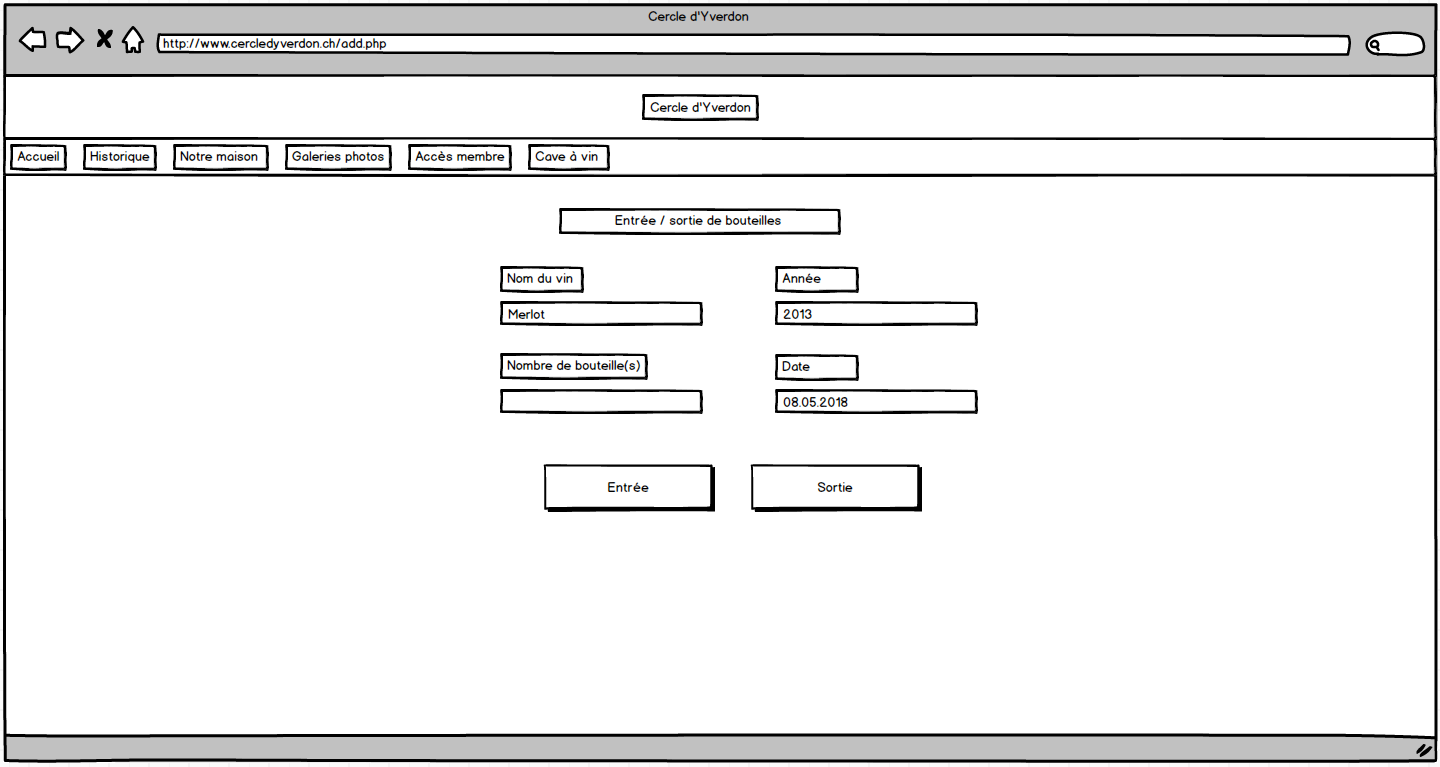
#### Ajouter un nouveau vin

Cette page permettra au caviste d’ajouter un nouveau vin pas encore présent en cave.



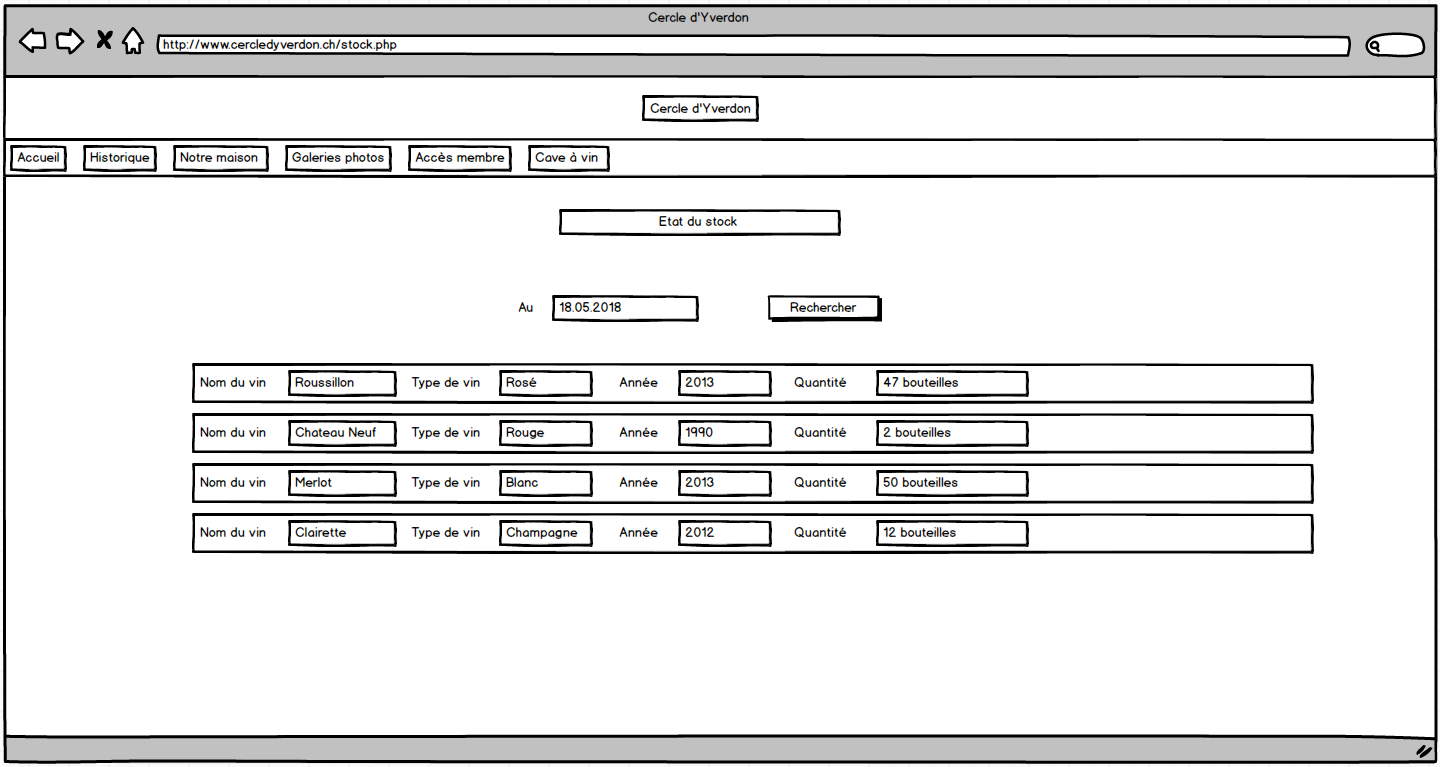
#### Entrée sortie du vin

Cette page permet au caviste d’ajouter les bouteilles qui rentrent dans la cave et mettre à jour dans la base de données. Il pourra aussi faire l’inverse, enlever le nombre de bouteilles qu’il enlève de la cave et mettre à jour dans la base de données.



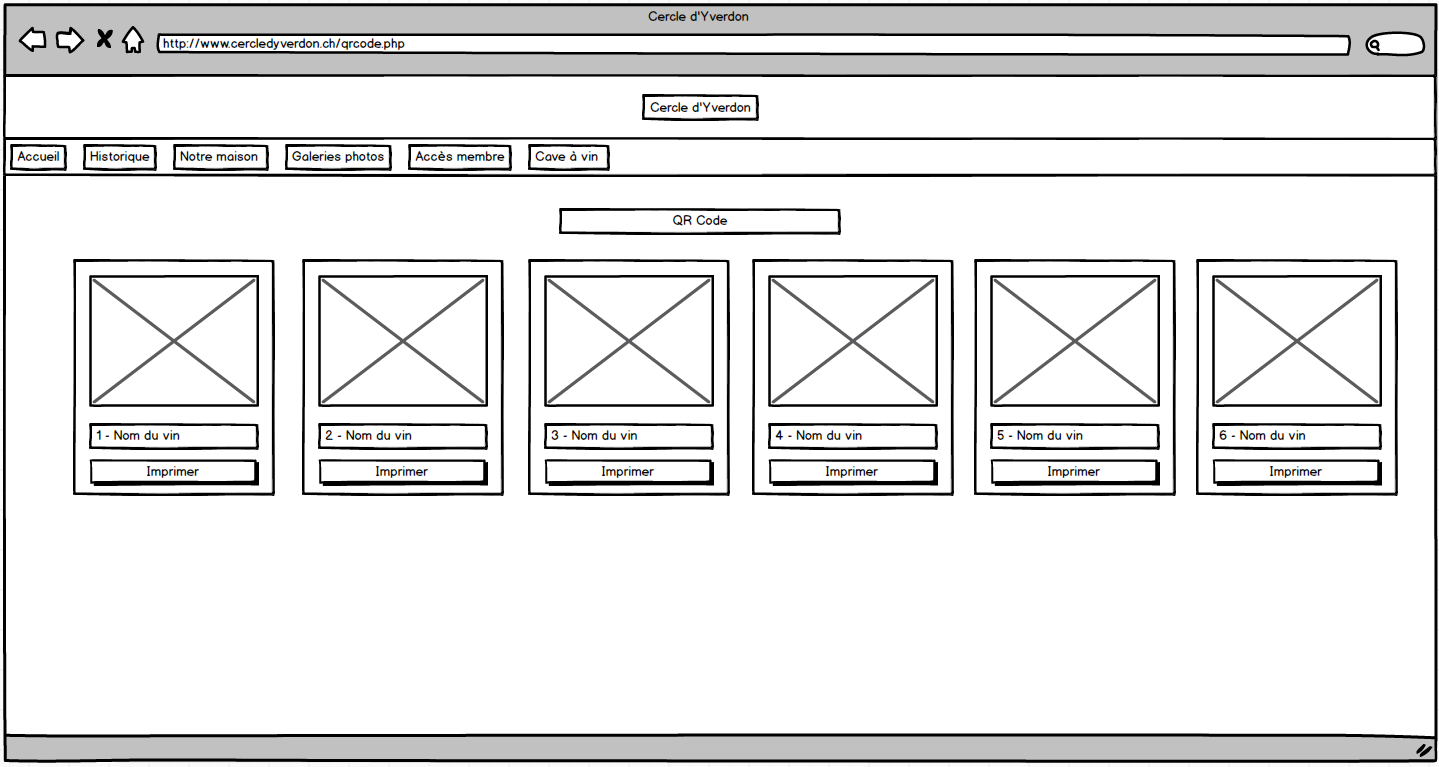
#### Aperçu du stock

Cette page permet au caviste d’avoir une vue d’ensemble de son stock de vin dans la base de données. Il aura aussi la possibilité d’avoir le stock à une date donnée.



#### Impression des QR Code

La liste des QR Code sera sur une page et le caviste aura la possibilité de les imprimer ou non.



### Use cases / Scénarios

#### Visualiser le stock

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste prend un smartphone et lance l’application du mobile. | L’application se charge et les menus s’affichent. |
| Il clique sur « stock » en bas à droite de l’application. | Les nouveaux menus s’affichent. |
| Il clique sur le bouton « scanner le QR code » | La caméra arrière s’affiche pour qu’il puisse scanner le QR Code. |
| Il vise le QR code avec son mobile. | Les informations du vin s’affichent sur l’application. |

#### Mettre à jour le stock d’un vin

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste prend un smartphone et lance l’application du mobile. | L’application se charge et les menus s’affichent. |
| Il clique sur « stock » en bas à droite de l’application. | Les nouveaux menus s’affichent. |
| Il clique sur le bouton « scanner le QR Code » | La caméra arrière s’affiche pour qu’il puisse scanner le QR Code. |
| Il vise le QR Code avec son mobile. | Les informations du vin s’affichent sur l’application. |
| Il remarque qu’il y a une erreur de quantité de bouteilles dans la base de données. Il entre la nouvelle valeur dans le champ « nouveau stock » et clique sur le bouton « appliquer ». | La valeur est alors enregistrée en local sur le téléphone. |
| Il clique sur le bouton « synchroniser » | Les valeurs sont ajoutées dans la base de données et mis à jour. |

#### Sortir des bouteilles du stock

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste prend un smartphone et lance l’application du mobile. | L’application se charge et les menus s’affichent. |
| Il clique sur le bouton « scanner le QR Code » | La caméra arrière s’affiche pour qu’il puisse scanner le QR Code. |
| Il vise le QR Code avec son mobile. | Les informations du vin s’affichent sur l’application. |
| Il entre le nombre de bouteilles qu’il sort de la cave dans le champ « sortie ». | Le champ est rempli par une valeur |
| Il signe son action en mettant son « login » dans un champ | Le champ est rempli par son login |
| Il clique sur « sortie » | Le mouvement est enregistré en mémoire sur le smartphone |
| L’utilisateur clique sur « synchroniser » | Le mouvement créé est envoyé à la base de données ce qui créera un mouvement. |

#### Entrer des bouteilles déjà existantes

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste prend un smartphone et lance l’application du mobile. | L’application se charge et les menus s’affichent. |
| Il clique sur « entrée/sortie » en bas à gauche de l’application. | Les nouveaux menus s’affichent. |
| Il clique sur le bouton « scanner le QR Code » | La caméra arrière s’affiche pour qu’il puisse scanner le QR Code. |
| Il vise le QR Code avec son mobile. | Les informations du vin s’affichent sur l’application. |
| Il entre le nombre de bouteilles qu’il ajoute dans sa cave dans le champ « entrée ». | Le champ n’est plus vide. |
| Il entre le fournisseur qui a fourni le vin dans le champs « fournisseurs ». | Le champ n’est plus vide. |
| Il signe son action en mettant son « login » dans un champ | Le champ est rempli par son login. |
| Il appuie sur le bouton « entrée ». | Le mouvement créé est envoyé à la base de données ce qui créera un mouvement. |
| L’utilisateur clique sur « synchroniser » | Le mouvement créé est envoyé à la base de données ce qui créera un mouvement. |

#### Ajout d’un vin avec des erreurs

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste prend un smartphone et lance l’application du mobile. | L’application se charge et les menus s’affichent. |
| Il clique sur « entrée/sortie » en bas à gauche de l’application. | Les nouveaux menus s’affichent. |
| Le caviste ne scanne pas de QR Code, n’entre pas le nombre de bouteilles et n’entre pas son login pour signer le mouvement | Aucune valeur n’est entrée |
| Un message d’erreur apparaît | L’utilisateur est informé des valeurs qu’il doit entrer pour faire un mouvement. |

#### Scanne un vin pas existant enregistré dans la mémoire du téléphone

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste prend un smartphone et lance l’application du mobile. | L’application se charge et les menus s’affichent. |
| Il clique sur « entrée/sortie » en bas à gauche de l’application. | Les nouveaux menus s’affichent. |
| Il clique sur le bouton « scanner le QR Code » | La caméra arrière s’affiche pour qu’il puisse scanner le QR Code. |
| Il vise le QR Code avec son mobile. | Aucune information ne s’affiche. |

#### Ajoute un nouveau vin dans la base de données

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique dans le menu sur « ajouter un vin ». | La page s’affiche. |
| Il ajoute le nom du vin, l’année, le type de vin, le nombre de bouteilles et le fournisseur. | Les champs sont remplis et ne sont plus vides. |
| Il clique sur le bouton « ajouter ». | Le vin est ajouté dans la base de données. |

#### Ajoute un nouveau vin en oubliant des champs

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique dans le menu sur « entrée sortie ». | La page s’affiche. |
| Il oublie d’entrer les informations dans un champ | Les champs sont remplis sauf ceux qui sont restés vides. |
| Il clique sur le bouton « ajouter ». | Un encadré rouge apparaît autour des champs que l’utilisateur a oubliés. |

#### Sortie d’un vin existant

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique sur « entrée sortie » pour sortir un vin. | La page s’affiche. |
| Il entre les informations du vin qu’il va prendre. | Les champs sont remplis et ne sont plus vides. |
| Il entre le nombre de bouteilles qu’il prend de la cave. | Le champ n’est plus vide. |
| Il clique sur le bouton « sortie » | Les données sont mise à jour dans la base de données. |

#### Sortie d’un vin existant en oubliant des champs

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique dans le menu sur « entrée sortie ». | La page s’affiche. |
| Il oublie d’entrer les informations dans un champ | Les champs sont remplis sauf ceux qui sont restés vides. |
| Il clique sur le bouton « sortie ». | Un encadré rouge apparaît autour des champs que l’utilisateur a oubliés. |

#### Entrée d’un vin existant

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique sur un sous menu de « cave à vin » pour sortir un vin. | La page s’affiche. |
| Il entre les informations du vin qu’il va prendre. | Les champs sont remplis et ne sont plus vides. |
| Il entre le nombre de bouteilles qu’il entre dans la cave. | Le champs n’est plus vides. |
| Il clique sur le bouton « entrée » | Les données sont mise à jour dans la base de données. |

#### Entrée d’un vin existant en oubliant des champs

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique dans le menu sur « entrée sortie ». | La page s’affiche. |
| Il oublie d’entrer les informations dans un champ | Les champs sont remplis sauf ceux qui sont restés vides. |
| Il clique sur le bouton « entrée ». | Un encadré rouge apparaît autour des champs que l’utilisateur a oubliés. |

#### Afficher le stock actuel

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique sur « stock » pour afficher le stock des vins. | La page s’affiche en listant les noms des vins, le type, l’année et la quantité présente dans la base de données. |

#### Afficher le stock en donnant deux dates

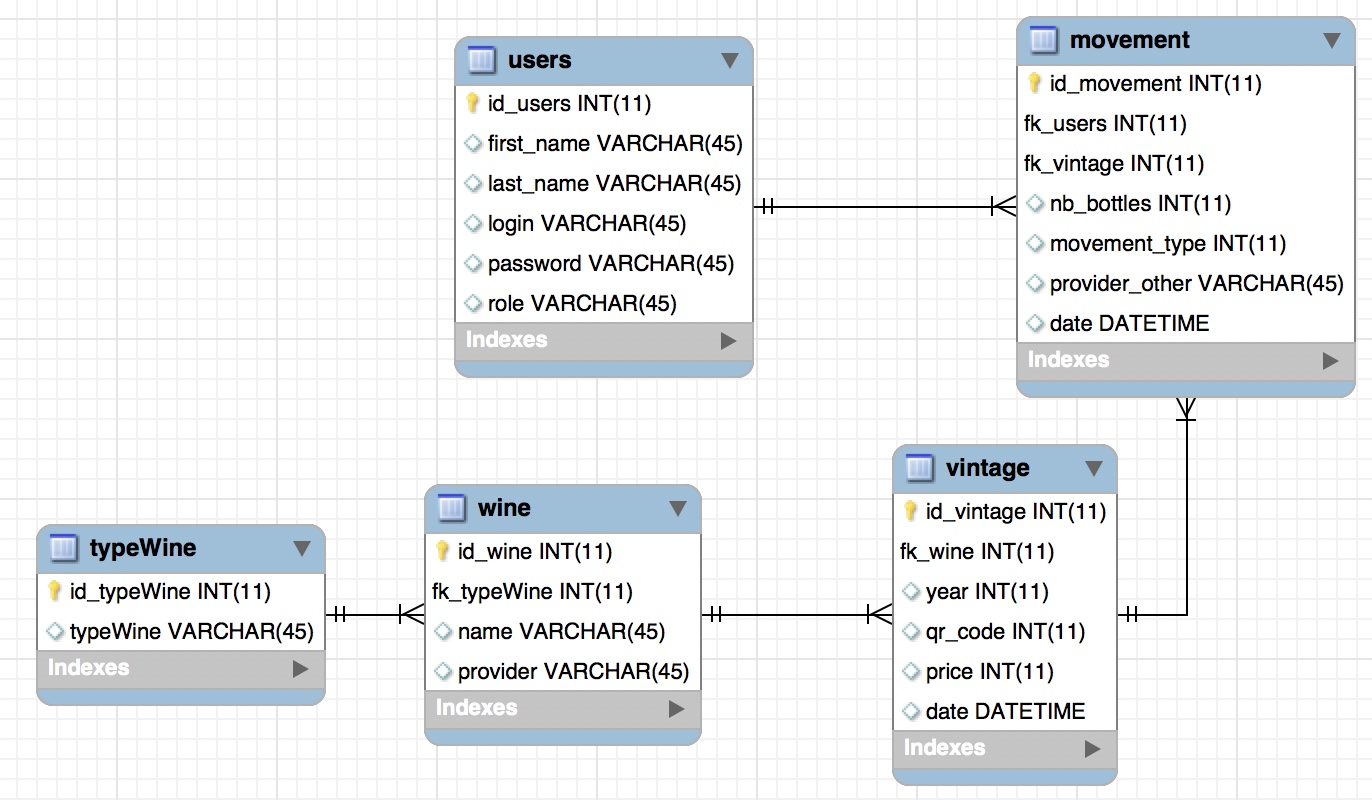
|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique sur « stock » pour afficher le stock des vins entre deux dates. | Le nouvelle page se charge. |
| Il entre deux dates entre laquelle il veut voir les stocks. | Les données se chargent. |

#### Impression des QR Code

|  |  |
| --- | --- |
| Action | Réaction |
| Le caviste va sur le site internet du cercle d’Yverdon. | Le site s’affiche. |
| Il clique dans le menu pour se connecter. | La nouvelle page s’affiche. |
| Il entre ses informations de connexion. | La connexion se fait. |
| Il clique sur « QR Code » pour afficher les QR Code. | Le nouvelle page se charge en listant tous les QR Code qui sont assignés à un vin. |
| Il clique sur le bouton « imprimer » en dessous du QR Code qu’il veut imprimer. | Le QR Code sélectionné s’affiche en plus grand. |
| Il clique à nouveau sur « imprimer » | Les informations d’impression s’affichent pour lancer l’impression. |

### MLD

Voici le MLD de la base de données.



Petite description de mon MLD, ci-dessus :

* La table « typewine »
  + Elle contiendra tous les types de vins que la cave pourra stocker, comme par exemple des bouteilles de blanc, de rouge, de mousseux, etc.
* La table « wine »
  + Cette table contiendra tous les noms des vins, comme par exemple un Merlot ou encore un Porto
  + Il y aura aussi un champ pour savoir qui a fourni le vin en question
* La table « vintage »
  + Elle servira à faire la différence entre les vins par rapport à leurs années. Si par exemple nous avons un Merlot de 2012 et 2013, dans cette table ça sera deux entités différentes même si elle provient du même vin.
  + Nous aurons la date du vin en question, pour pouvoir les différencier entre un Merlot de 2012 et un autre de 2013, comme expliqué juste en dessus
  + Nous aurons un QR Code pour la partie mobile afin de savoir à quel vin on a affaire lorsque l’on scanne le QR.

* La table « movement »
  + Cette table servira à lister tous les mouvements qui seront faits dans la cave. Si par exemple nous voulons sortir 6 bouteilles d’un vin quelconque, et bien le mouvement sera inscrit dans cette table.
  + Pareil si nous approvisionnons un vin déjà en cave, ça fera une entrée en cave
  + Le champs « movement\_type » sert à savoir quelle action a été effectuée. Dans ce champ, il n’y aura que ces trois valeurs suivantes : « -1 », « 0 » et « 1 ». La valeur « -1 », sert à informer que l’action qui a été faite était de sortir un vin. Ensuite, le « 1 » sert à faire une nouvelle entrée en cave. Pour finir, le « 0 » sert à faire l’inventaire d’un vin souhaité.
  + La quantité d’un vin restant est calculée de la manière suivante. En premier lieu on recherche dans la base de données le dernier « movement\_type » qui est égal à « 0 ». Ensuite, on parcourt tous les mouvements qui se sont passés après le dernier inventaire et on additionne tout ensemble.
* La table « users »
  + Pour finir cette table servira à savoir qui a effectué un mouvement dans la cave
  + Elle permettra aussi aux utilisateurs de se connecter

# Réalisation

## Site web

Je vous ai mis, ci-dessous des captures d’écrans qui permettront de vous expliquer comment le site fonctionne.

### Nouvelle entrée en cave

Cette page permet au caviste d’ajouter un nouveau vin dans la cave. Pendant la période d’analyse, j’ai créé des maquettes de mes pages pour le site et de l’application mobile que j’ai été montrer au client. Il les a validés, c’est pour cela que les champs de la capture d’écran ci-dessus sont présents sur la page. De plus, s’il appuie sur le bouton ajouter sans valeur dans tous les champs, les champs deviennent rouges.

Une fois que toutes les informations du vin sont entrés, il clique sur le bouton ajouter ce qui ajoutera ce vin dans la base de données. Pour le type de vin, c’est le client lui-même qui m’a donné tous les types de vin qui pouvait être un jour en cave.

Voici la liste qu’il m’a fournie :

* Vins blancs
* Vins rosés
* Vins rouges
* Vins mousseux/Champagnes
* Vins liquoreux

## Entrée, sortie de bouteilles

Voilà à quoi ressemble la page qui permet de faire entrer ou sortir un vin présent en cave. Le caviste sélectionne dans la liste déroulante, le vin qu’il va entrer ou sortir et l’année du vin apparaitra.



Pour l’exemple, j’ai sélectionné le « Merlot » dans la liste déroulante et l’année du vin est apparue à droite. Il peut y avoir plusieurs années pour un vin choisi, au caviste de sélectionner l’année correspondant au vin qu’il entre ou sort. Suivant l’action qu’il veut faire il clique sur les boutons « Entrée », « Sortie » en bas.

## Stock

La page qui résume le stock dans la base de données se constitue comme ça. Il liste tous les éléments de la base de données sur la page lorsqu’on y arrive dessus.

Il affiche un tableau avec le nom du vin, le type de vin que c’est, l’année et enfin le stock présent dans la base de données.



Lorsqu’on entre une date et qu’on appuie sur le bouton rechercher, il va sélectionner le dernier inventaire dans la base de données avant la date entrée. Il va ensuite afficher la quantité des vins restants.

Il faudrait encore vérifier cette fonction parce que les chiffres se changent mais je ne suis pas sûr que les valeurs qui s’affichent soient les bonnes.

## Application mobile

Ci-dessous vous trouverez les pages principales permettant de faire des actions vers la base de données sur le serveur.

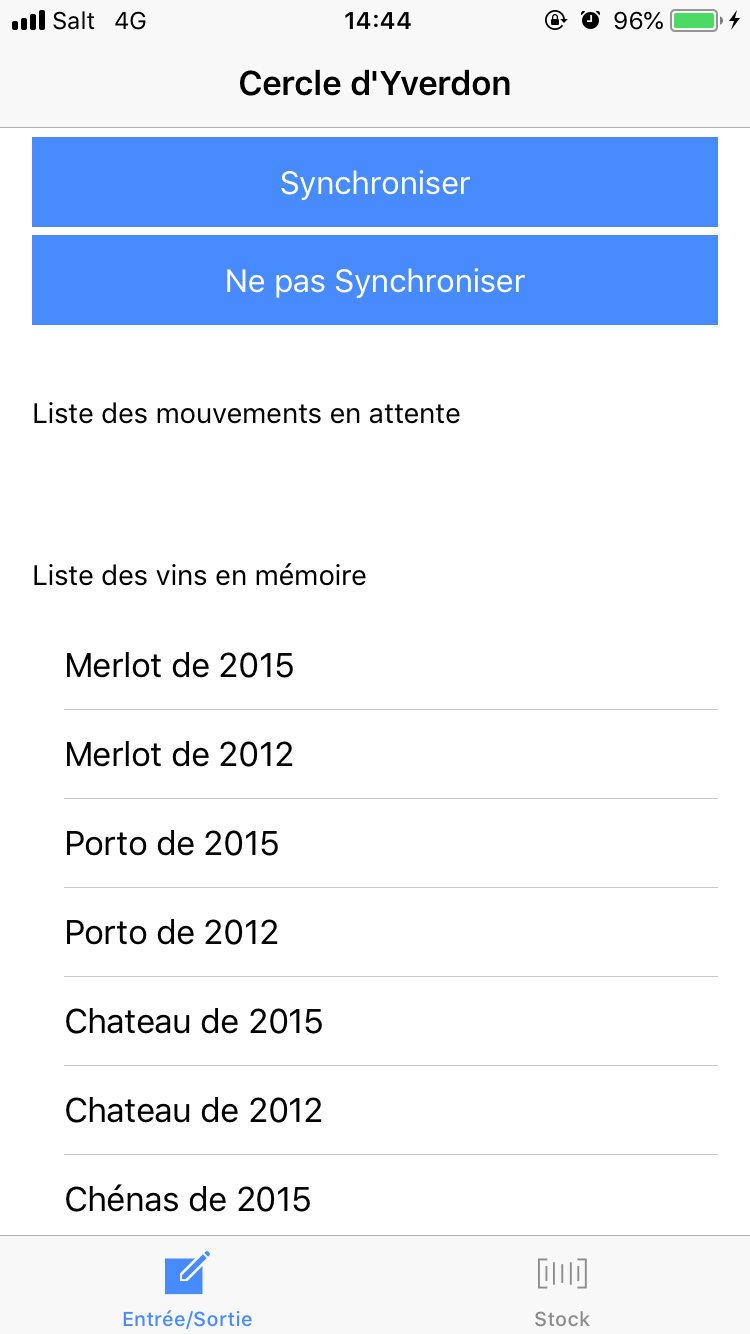
### Entrée / sortie de bouteille



Dès que l’on ouvre l’application la première page qui s’affiche c’est celle-ci. Sur la page d’accueil nous trouverons six boutons qui permettront de faire des actions. Le premier bouton, « scanner le QR Code » permet de scanner un QR Code pour que les données du vin soient renvoyées et affichées dans l’application.

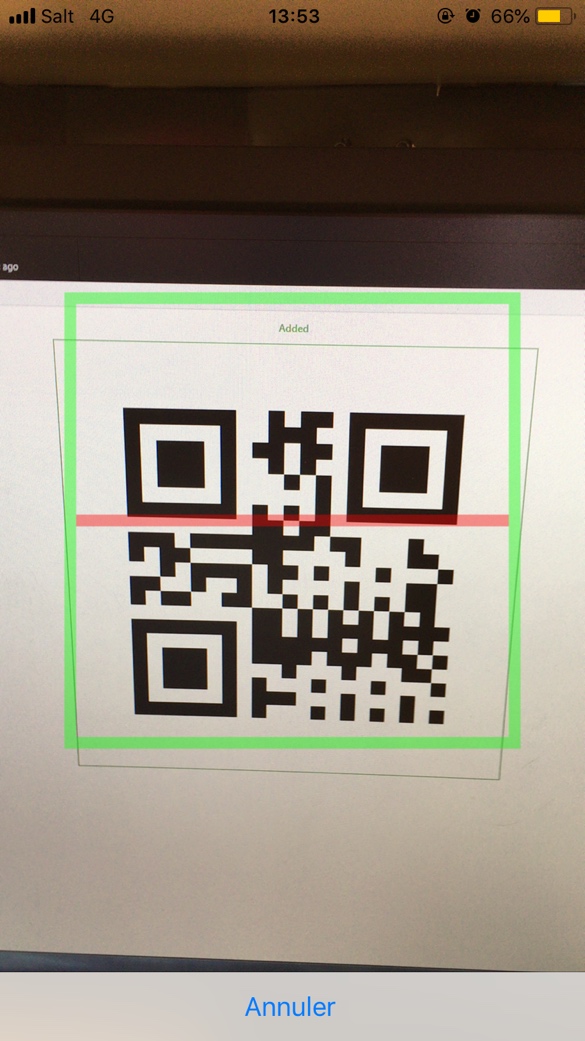
Les boutons « entrée » et « sortie » permettront au caviste de faire des mouvements, c’est-à-dire, à chaque fois qu’il entre ou qu’il sort une bouteille de sa cave, il appuie sur le bouton qui correspond à l’action qu’il a entreprise.

Le bouton « inventaire » permet au caviste de faire l’inventaire à un moment donné en ajustant le stock.

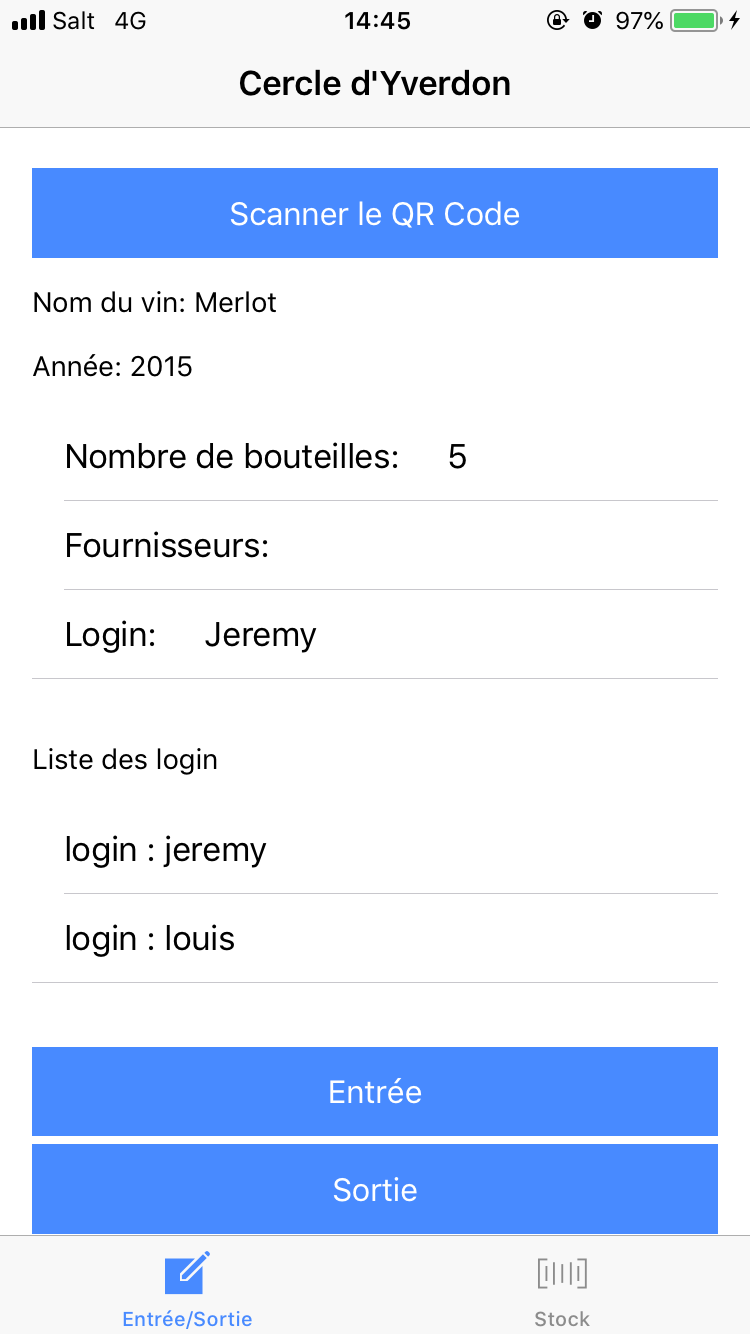
Le bouton « synchroniser » permettra à l’utilisateur d’envoyer les mouvements qui sont enregistrés sur le mémoire du téléphone.

Le bouton « ne pas synchroniser » permet au caviste de supprimer tous les mouvements en attente s’il s’aperçoit qu’il a fait une erreur sur l’un d’eux.

La liste des vins est aussi enregistrée sur la mémoire et est affichée en bas de page.



Voilà la page qui s’affiche lorsque nous cliquons sur « scanner un QR Code ». L’appareil photo arrière du portable s’ouvre permettant de scanner ce qu’on le souhaite.



Lorsque le scan du QR Code est concluant, les champs « Nom du vin » et « Année », en haut de page, se remplissent avec les valeurs du vin souhaité. Le caviste, dès qu’il a entré des valeurs dans les champs, pourra entrer ou sortir des bouteilles de la cave, en signant avec son login présent.

La liste des logins est affichée en dessous pour savoir lequel ils peuvent mettre afin de signer le mouvement.

Une fois les champs remplis, il appuie sur « entrée » ou « sortie » pour créer le mouvement.



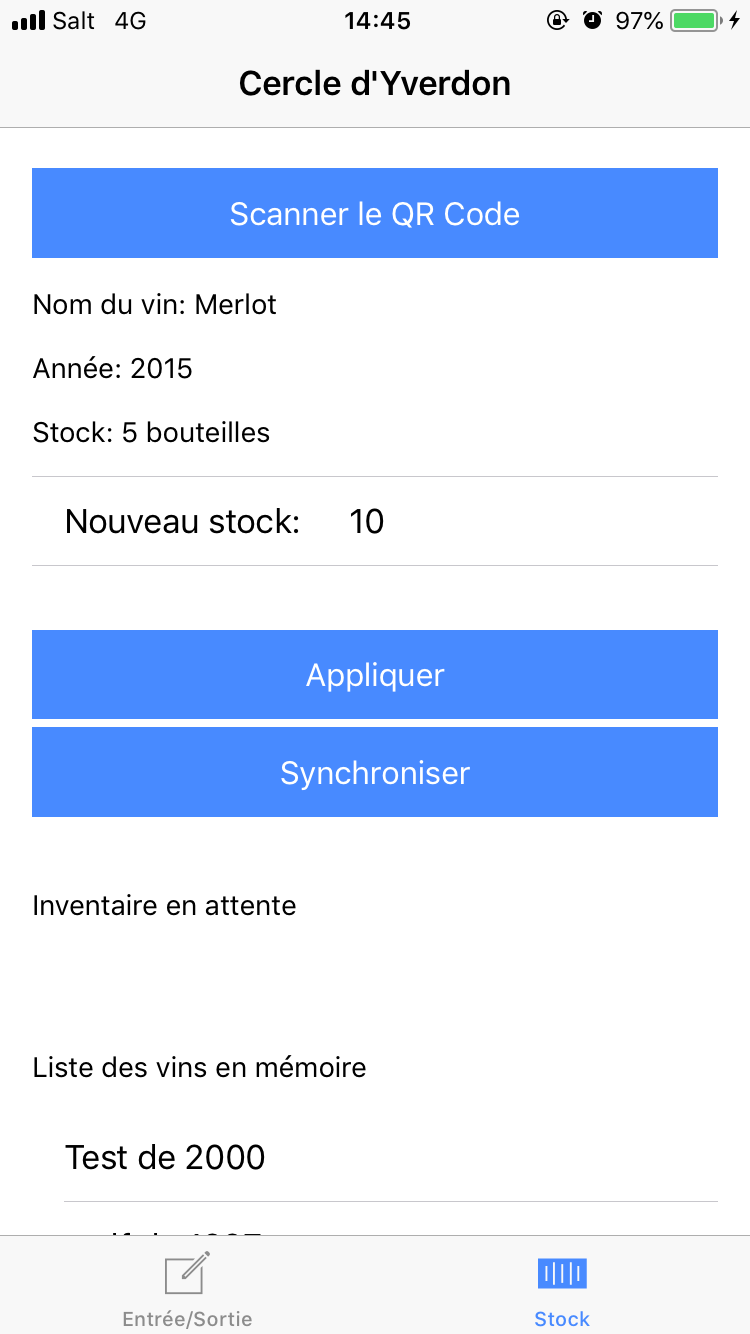
Une fois que le mouvement a été créé, il est listé en dessous en disant qu’il est en attente. À cette étape, rien n’a été envoyé à la base de données.



Une fois que le caviste a appuyé sur le bouton « synchroniser » tous les mouvements enregistrés en local sur le téléphone sont envoyés à la base de données.

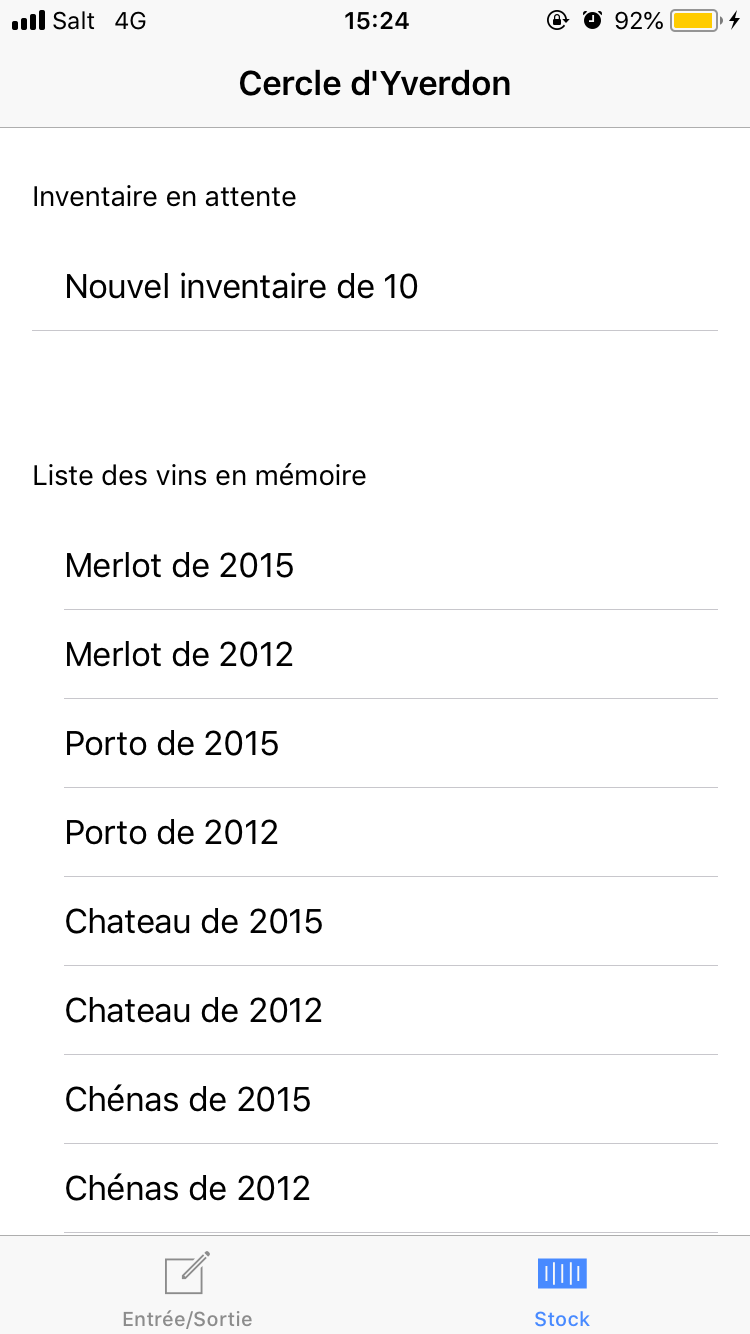
Si le message « syncro ok » en bas de page s’affiche, c’est que les requêtes se sont correctement déroulés et que les mouvements en attente se sont supprimés du stockage du téléphone.

### Gérer le stock

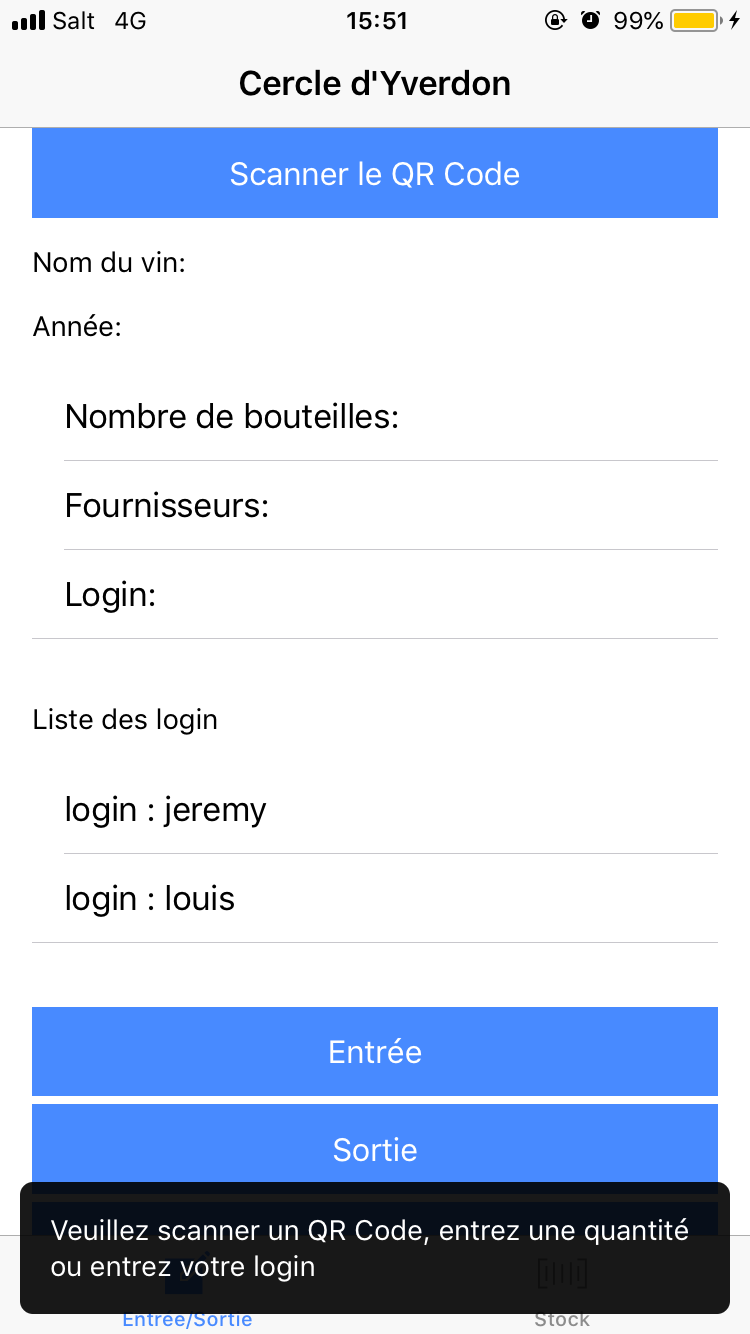


La page pour gérer le stock fonctionne de la même manière que les captures d’écrans précédentes. Une fois que le QR Code a été scanné les informations du vin s’affichent.

Ensuite de ça, s’il voit qu’il y a une erreur du nombre de bouteilles dans la base de données en comparant ce qu’il a en cave, il peut ajuster le stock



Le nouvel inventaire s’affiche ensuite comme cela. Dès que le caviste clique sur « synchroniser » les inventaires sont envoyés à la base de données ce qui mettra à jour la quantité des vins.



Si l’utilisateur oublie de scanner un QR Code, d’entrer une quantité ou bien son login lors du mouvement, un message d’erreur s’affiche en lui indiquant ce qu’il doit faire pour créer son mouvement.

L’application doit pouvoir fonctionner en mode hors connexion, nous avons décidé avec M. Carrel, de sauvegarder les données en local sur le téléphone, on sauvegardera tous les mouvements ainsi que les inventaires qui seront faits par l’utilisateur, lorsqu’il cliquera sur le bouton « synchroniser » il enverra les données en stock sur le smartphone vers la base de données et enregistrera aussi le nouveau stock en mémoire.

Cette solution pourra dans le futur être amélioré.

## Dossier de réalisation

Tous les fichiers pour le site et l’application mobile sont sauvegardés sur un répertoire GitHub. À chaque nouvelle fonction, modification ou suppression sur un fichier, je sauvegardais mes fichiers sur mon profil sur GitHub.

Les fichiers étaient enregistrés sur le disque C: de la machine au CPNV dans le dossier wamp puis dans www. J’ai ensuite créé des dossiers me permettant de différencier mon application mobile et le site.

### Partie site

Dans le dossier du site nous avons plusieurs dossiers qui sont : controller, model, phpqrcode, public, qr\_code et le dossier view.

Le seul fichier présent dans le dossier « controller » contient toute la logique des actions que l’utilisateur effectuera.

Dans le dossier « model » il y a aussi un seul fichier. Ce dernier contient toutes les requêtes qui sont envoyées à ma base de données. Le résultat est ensuite renvoyé au fichier qui se trouve dans le dossier « controller ».

Mon dossier « view », lui, contient toutes les pages qui sont affichées sur le site. Dans ces fichiers, il y a uniquement des balises HTML et très peu de PHP.

Le dossier « public » lui sert à contenir tous les fichiers CSS et JavaScript.

Le dossier « qr\_code », contient toute les images des QR Code qui sont générés lors de la création d’un nouveau vin sur le site.

Pour finir, le dossier « phpqrcode », est le dossier qui me permet de générer les QR Code avec une fonction en PHP. Sans ce dossier je n’aurai pas pu les générer.

À la racine, du dossier du site, il y a d’autres fichiers. L’ « index.php » me sert à diriger l’utilisateur dans les fonctions du site. « searchWine.php » me permet d’afficher les années d’un vin sélectionné sur le site. « inout.php » est utilisé pour l’application mobile. Elle me permet d’ajouter ou retirer des bouteilles. « update.php » est aussi utilisé par l’application mobile. Elle permet uniquement de mettre à jour le stock dans la base de données. Pour finir, « stock.php » me sert à stocker tous les vins dans la base de données sur la mémoire du smartphone.

### Partie application mobile

Lors de la création d’un nouveau projet sur ionic il crée un dossier avec tous les fichiers que l’application aura besoin pour fonctionner. Les fichiers qui seront énumérés plus tard sont ceux où j’ai ajouté des choses afin de pouvoir utiliser des fonctions sur d’autres pages, ou celle sur lesquelles j’ai fait le développement.

Le dossier « node\_modules » contient tous les modules qu’ionic est capable de comprendre. Par exemple pour mon projet, on y retrouve les modules qui me permettent de scanner des QR Codes ainsi que d’enregistrer des informations en local sur le téléphone.

Dans le fichier « app.module.ts » se trouvant dans les dossiers « src » puis dans « app », lui contient, les importations des modules qui seront utilisé une fois ou l’autre par l’application.

Ensuite, dans le dossier « pages », se trouvant dans « src », contiennent deux dossiers où sont développés les pages de l’application. Le dossier « home » contient la page permettant de faire les entrées et sorties des bouteilles de vin et le dossier « about » permet de faire les inventaires pour un vin. Dans chacun de ces dossiers il y a un fichier .html qui contient l’affichage de l’application et un autre fichier .ts qui lui contient toutes les fonctions et développement des pages.

Le dossier « tabs » permet de faire le lien entre les deux pages avec un petit menu en bas de page de l’application.

## Description des tests effectués

Les tests qui ont été effectué se sont déroulés durant tout le projet. Lorsqu’une fonction a été implémentée je l’ai testée de plusieurs manières différentes. J’ai fait cette stratégie autant pour l’application que pour le site.

M. Chavey a pu tester l’application et je l’ai amélioré avec les erreurs qu’il avait trouvée.

Malheureusement je ne peux pas faire de tableau parce que je ne connais pas les dates de quand les tests ont été effectués. De plus avec la modification de la base de données qui a été faite une semaine avant la fin du TPI, je n’ai pas pu faire beaucoup de tests.

## Erreurs restantes

Les erreurs restantes se situent sur l’application mobile, lors d’un mouvement l’utilisateur doit entrer son login pour signer son mouvement. J’ai affiché la liste des logins présents dans la base de données pour réduire le risque d’erreur, mais s’il entre n’importe quoi dans le champ la requête ne s’exécute pas et le mouvement n’est pas créé dans la base de données.

Pour réparer cette erreur je dirai qu’il faudrait faire un formulaire d’authentification lorsqu’on lance l’application.

Sur la page stock, la fonction permettant d’afficher le stock avec une date donnée fonctionne mais je ne suis pas sûr que les valeurs soient justes.

Toujours sur la page stock, lorsqu’on clique sur le bouton rechercher sans avoir entré de date il y a des erreurs, il faut recharger la page en cliquent sur « stock » dans le menu. Par contre si on sélectionne une date et ensuite on clique sur rechercher il n’y a pas d’erreurs.

## Liste des documents fournis

Voici la liste ci-dessous des documents qui seront fournis en plus de la documentation :

* Le Gantt initiale et la mise à jour
* Le journal de travail
* Un fichier complet qui contiendra
  + Le site et son développement
  + La base de données et un script pour la création de celle-ci
* Un autre fichier qui contiendra
  + Tout le développement de l’application mobile

# Conclusions

Tout d’abord, grâce à ce TPI, j’ai pu me perfectionner avec le Framework ionic, en créant de nouvelles fonctionnalités que je n’avais pas eu la chance de voir durant mon pré-TPI.

Pour les points positifs, j’ai bien aimé le fait d’avoir un client qui demande qu’on lui fournisse une application et des pages internet pour la gestion de sa cave, parce que je sais que je n’ai pas effectué ce projet pour rien, parce qu’il va l’utiliser. Ce n’est pas un projet que j’ai effectué, entre guillemets, pour rien.

Le fait d’avoir du développement en TPI aussi est un des points positifs, puise que c’est dans ce domaine que je me sens le plus à l’aise.

Pour les points négatifs, c’est aussi le fait d’avoir un client, puisqu’il y a des attentes et un stresse plus présent, parce qu’on se dit que tous les objectifs doivent être atteints et qu’on doit satisfaire le mandant.

Pour les difficultés particulières. J’ai eu un peu de peine à réaliser des fonctions sur l’application mobile, mais grâce à l’aide de la documentation d’ionic, à une vidéo YouTube ainsi que de l’aide qui m’a été fournie par M. Carrel, j’ai réussi à développer l’application.

Pour les suites possibles du projet, je dirai que ce serait d’améliorer l’application mobile en mettant un petit formulaire de connexion lors du lancement de celle-ci. Cela permettra de faciliter certaines requêtes qui sont envoyées à la base de données.

Ensuite pour la partie web, la fonction permettant de voir la quantité en donnant une date doit être améliorée, comme je l’ai dit dans un chapitre précédent je ne suis pas sûr à 100% qu’elle soit correct.

## Conclusion personnelle

Je pense avoir bien géré les objectifs qui m’étaient demandés. Par contre le temps de travail qui était prévu je ne l’ai pas du tout respecté puisque j’ai passé plus d’heure sur l’implémentation que ce qui était prévu dans le cahier des charges.

Pour ma défense les petits problèmes et le changement lié à la base de données ne m’ont pas facilité la tâche pour respecter les heures données dans le cahier des charges.

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

### Situation de départ

Le cercle d’Yverdon est venu vers le CPNV pour demander que quelqu’un fasse une application pour que la personne responsable de la cave à vins puisse la gérer de manière simple et efficace. J’ai eu la tâche pour ce TPI de réaliser une application mobile qui sera capable de fonctionner sans connexion. Une partie web sera aussi implémentée avec les mêmes fonctions que sur l’application. Une seule fonction ne sera pas présente sur l’application, ce sera celle qui permet d’ajouter un nouveau vin en cave qui n’y a jamais été auparavant.

### Mise en œuvre

Pour l’application mobile, j’ai repris une fonction que j’avais développée pendant mon Pré-TPI, celle de scanner un QR Code est de lire sa valeur. J’ai dû créer les fonctions me permettant d’envoyer des données à la base de données et celle qui permet à l’application de fonctionner en mode hors connexion.

Une fois que les fonctions les plus simples de l’application étaient implémentées, je me suis attardé à la partie web. J’ai commencé par créer plusieurs pages en mettant mes balises HTML pour que mes pages ressemblent le plus possible aux maquettes réalisées ultérieurement. Une fois que toutes mes balises avaient été placées je me suis attardé à la partie fonction en PHP. J’ai fait mes fonctions comme on m’a enseigné durant les modules de PHP au CPNV. Dans un de mes critères de mon cahier des charges étaient que le site devait avoir une architecture en MVC, ayant de vague notion je me suis référé à des tutoriels sur internet pour passer tout mon site avec cette architecture.

Lorsque j’avais des questions j’ai pu les poser à un de mes camarades de classe (Alexandre Junod) qui lui code aussi son site avec l’architecture qui m’était demandée.

### Résultats

Pour la partie du site, j’ai repris l’affichage présent sur le site du cercle d’Yverdon et je l’ai mis sur mes pages que j’ai créés. Grâce à cela, elles s’adaptent au format des écrans sur lequel le site est visualisé. Tous les objectifs ont été réalisé.

Pour la partie mobile, j’ai créé deux pages qui me permettent de faire des actions distinctes. Sur une des deux pages, je peux créer des mouvements, ce qui revient à sortir ou entrer des bouteilles de la cave. Sur l’autre je peux ajuster le stock d’un vin si la quantité en cave ne correspond pas à celle de la base de données.

## Glossaire

|  |  |
| --- | --- |
| Framework | Un Framework est une structure qui sert à créer les fondations ainsi que les grandes lignes de tout ou d’une partie d’un logiciel. |
| GitHub | GitHub est un service web d’hébergement et de gestion de logiciel utilisant le logiciel de gestion de version Git. |

## Sources – Bibliographie

Vidéo YouTube sur laquelle je me suis aidée

* <https://www.youtube.com/watch?v=1v-BbgFM5ps>

Site sur lequel je me suis aidé pour faire la fonction de l’Ajax

* <https://www.w3schools.com/php/php_ajax_database.asp>

Site sur lequel j’ai trouvé comment générer les QR Code et où trouver les fichiers à télécharger pour les importer ensuite

* <http://phpqrcode.sourceforge.net/>

Définition de Framework depuis Wikipédia :

* <https://fr.wikipedia.org/wiki/Framework>

Définition de GitHub depuis Wikipédia :

* <https://fr.wikipedia.org/wiki/GitHub>

## Journal de bord

|  |  |
| --- | --- |
| **Date** | **Événement** |
| 09.05.2018 | Faire le point avec M. Chavey pour la partie analyse, je lui ai montré mes maquettes ainsi que mon MLD et mon MCD. On a fait quelques modifications pour la base de données. |
| 14.05.2018 | J’ai pris rendez-vous avec le client pour lui montrer les maquettes que j’avais créées pour voir avec lui si elles lui convenaient. Dans l’ensemble, il était content, il m’a quand même suggéré de faire quelques modifications. |
| 23.03.2018 | Visite de l’expert n°2, M. Laurent Tuchat pour parler de l’avancement du projet. |
| 01.06.2018 | Changement dans la base de données suite à la demande de M. Carrel. Nous avons remplacé deux champs par deux autres pour différencier les mouvements qui sont faits. |

## Manuel d'Installation

Voici les étapes à faire pour pouvoir utiliser les pages du site internet ainsi que l’application mobile.

### Windows

* Installer WampServer sur l’ordinateur, simule un serveur local et permet l’utilisation de PHP.
* Lancer WampServer.
* Ajouter ensuite le dossier de mon site à cette adresse sur votre ordinateur : C:/wamp/www/
* Installer MySQL Workbench, permet de gérer une base de données.
* Lancer le ensuite et importer le fichier SQL de création de la base de données, qui se trouve là : C:/wamp/www/TPI/Base de données/createDataBase.sql.
* Lancer ensuite un navigateur et aller à l’adresse : localhost/TPI/site/

### Mac

* Installer MampServer sur l’ordinateur, lui aussi simule un serveur local pour l’utilisation de PHP.
* Lancer MampServer.
* Ajouter le dossier du projet à cette adresse : /Applications/MAMP/htdocs/
* Installer MySQL Workbench, permet de gérer une base de données.
* Lancer l’application et importer ensuite le fichier SQL de création de la base de données, qui se trouve là : /Applications/MAMP/htdocs/TPI/Base\_de\_données/createDataBase.sql
* Ouvrir la page qui se trouve dans /Applications/MAMP/htdocs/TPI/site/model/frontend.php – à la ligne N°9 rajouter entre les guillemets « root »
* Se rendre sur un navigateur et se rendre à l’adresse suivant : localhost:8888/TPI/site/

## Aides externes

**Alexandre Junod :** Il m’a aidé à passer une page avec l’architecture MVC. Suite à cela je me suis débrouillé pour faire le changement sur les autres pages.

**M. Xavier Carrel :** Il m’a surtout aidé pour la partie mobile avec les nouvelles fonctions à développer que je n’avais pas vu durant le Pré-TPI.

**M. Jean-Philippe Chavey :** Étant mon chef de projet lors du TPI, il est venu me voir régulièrement pour voir l’avancée du projet. Il m’a aussi donné un coup au début du TPI pour concevoir ma base de données.