Sploks est un programme pour gérer la location de matériel de sports d'hiver

Sploks

Projet TPI 4ème année

Auteur : Marwan Alhelo

E-mail : [marwan.alhlo@cpnv.ch](mailto:marwan.alhlo@cpnv.ch)

Classe : SI-C4a

Date :31.05.2022

Table des matières

[1 Analyse préliminaire 2](#_Toc104868050)

[1.1 Introduction 2](#_Toc104868051)

[2 Objectifs 2](#_Toc104868052)

[2.1.1 Sprint 1 : Gérer le carnet d’adresse 2](#_Toc104868053)

[2.1.2 Sprint 2 : Historique et gérer le lien entre deux groupe et familles 2](#_Toc104868054)

[2.1.3 Sprint 3 : Générer un e-mail de rappel 2](#_Toc104868055)

[3 Planification initiale 2](#_Toc104868056)

[4 Analyse / Conception 2](#_Toc104868057)

[4.1 Concept 2](#_Toc104868058)

[4.2 Modèle conceptuel de données (MCD) 3](#_Toc104868059)

[4.3 Analyse des risques technique 4](#_Toc104868060)

[4.4 Modèle logique des données (MLD) 5](#_Toc104868061)

[4.5 Maquettes 6](#_Toc104868062)

[4.6 Stratégie de test 10](#_Toc104868063)

[4.7 Dossier de conception 11](#_Toc104868064)

[5 Réalisation 13](#_Toc104868065)

[5.1 Dossier de réalisation 13](#_Toc104868066)

[Description des tests effectués 15](#_Toc104868067)

[5.2 Erreurs restantes 18](#_Toc104868068)

[5.3 Liste des documents fournis 18](#_Toc104868069)

[6 Conclusions 18](#_Toc104868070)

[7 Annexes 19](#_Toc104868071)

[7.1 Résumé du rapport du TPI / version succincte de la documentation 19](#_Toc104868072)

[7.2 Sources – Bibliographie 19](#_Toc104868073)

[7.3 Journal de travail 19](#_Toc104868074)

[7.4 19](#_Toc104868075)

[7.5 Manuel d'Installation 19](#_Toc104868076)

[7.6 Manuel d'Utilisation 19](#_Toc104868077)

[7.7 Archives du projet 19](#_Toc104868078)

# Analyse préliminaire

## Introduction

Sploks est un potentiel programme de remplacement de Coliks, qui est utilisée depuis 17 ans dans le magasin Sports-Time d'Echallens pour gérer la location de matériel de sports d'hiver.

Sploks permet facilement de louer du matériel de sports d’hiver, générer et stocker des contrats de location, ainsi que tenir les informations du matériel en stock à jour

J’ai choisi car c’est un travail qui m’attire la gestion de développement et la langage python et puis c’est pour un vrai client un magasin à Echallens en cas le programme fonctionne comme prévu car c’est une mise à jour d’un ancien programme c’est un défi qu’on aura dans la vie réelle.

# Objectifs

Les objectifs seront structurés par de sprint et chaque sprint contiens de stories et chaque story contient de taches et chaque tâche contient de tests

La planification initiale incarne les objectifs au démarrage du projet,

### Sprint 1 : Gérer le carnet d’adresse

En tant que propriétaire du magasin, je veux maintenir un carnet d’adresse clients à jour.

Je veux conserver les informations suivantes sur un client :

• Prénom

• Nom

• Adresse

• Adresse mail (optionnel)

• Numéro de téléphone

Il est tout-à-fait plausible que deux clients aient le même prénom/nom. Dans ce cas, je les distingue par leur numéro de téléphone.

Je veux pouvoir :

• Ajouter un nouveau client à ma liste.

• Modifier n’importe quelle information concernant un client.

• Désactiver un client, mais pas l’effacer (car je veux garder les traces)

• Afficher une liste de clients actif et/ou inactif et filtrer cette liste pour retrouver une personne spécifique.

En plus des informations de base, la liste me montrera pour chaque client :

• Le chiffre d’affaires (en matière de location) réalisé avec ce client

• S’il a du matériel à retourner en retard

### Sprint 2 : Historique et gérer le lien entre deux groupe et familles

Pour chaque client, je veux pouvoir maintenir et consulter un historique dans lequel figure :

• La date d’arrivée

• Les changements de données (adresses, numéro de téléphone, email,)

• Les changements de liens

• Les emails de rappels

• Annotation libre

• Date(s) de changement d’état (activé/désactivé)

Gérer le lien entre deux groupe et familles :

* Lier des clients entre eux (familles, groupes).

### Sprint 3 : Générer un e-mail de rappel

À partir de la fiche descriptive d’un client qui a du matériel à retourner en retard, je peux générer un e-mail de rappel.

# Planification initiale

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprints** | | | | | | | | | |
| **Gérer le carnet d’adresse** | | | |  | **Historique** | |  | **Générer un email de rappel** | |
| **Tâches** | | | |  | **Tâches** | |  | **Tâches** | |
| **1. Saisir les informations d’un nouveau client** | **2. Modifier les informations** | **3. Désactiver un client** | **4. Parcourir le carnet d’adresse** |  | **1. Consulter l’historique** | **2.Gérer des liens entre clients** |  | **1. Générer un email de rappel** | |
| **Date** | | | |  | **Date** | |  | **Date** | |
| **Début** | | | **Fin** |  | **Début** | **Fin** |  | **Début** | **Fin** |
| **02.05.2022** | | | **12.25.2022** |  | **13.05.2022** | **24.05.2022** |  | **30.05.2022** | **31.05.2022** |
| **Sprint Review** | | | |  | **Sprint Review** | |  | **Sprint Review** | |
| **12.05.2022 À 11h30** | | | |  | **24.05.2022 À 15h20** | |  | **31.05.2022 À 11h30** | |
| **Heure** | | | **Percentage** |  | **Heure** | **Percentage** |  | **Heure** | **Percentage** |
| **Documentation** | | | |  | **Documentation** | |  | **Documentation** | |
| 11.0 | | | 20% |  | 6.0 | 20% |  | 1.20 | 20% |
| **Implémentation** | | | |  | **Implémentation** | |  | **Implémentation** | |
| 27.0 | | | 50% |  | 15.0 | 50% |  | 3.00 | 50% |
| **Analyse** | | | |  | **Analyse** | |  | **Analyse** | |
| 8.0 | | | 15% |  | 4.5 | 15% |  | 0.58 | 15% |
| **Test** | | | |  | **Test** | |  | **Test** | |
| 8.0 | | | 15% |  | 4.5 | 15% |  | 0.58 | 15% |
| **Total** | | | |  | **Total** | |  | **Total** | |
| 54.0 | | | 100% |  | 30.0 | 100% |  | 5.36 | 100% |

# Analyse / Conception

## Concept

* Le schéma du réseau au magasin de sports et Sploks va fonctionner dans cet enivrement
* Ce programme destiné au personnel du magasin et ils sont cinq personnes avec de connaissance de base en informatique
* Pour sploks soit opérationnel il faut ajouter un server LAMP pour la base de onné
* Les postes ce sont de ordinateurs avec windows 10 comme system

D’exploitation



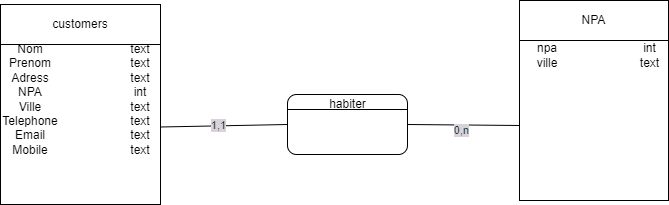
## Modèle conceptuel de données (MCD)

Pour une base de données permet d'identifier les principales entités à représenter, leurs relations et leurs attributs, et d'analyser la structure conceptuelle du système d'information. Contrairement à un modèle de données logique (MLD) ou à un modèle de données physique (MPD), un MCD est plus abstrait.

Ces deux tables que je suis en train de développer mes fonctionnalités comme c’est le cas de figure la table NPA montre dans une ville on peut trouver plusieurs clients qui habitent ou bien personne

La tables Customers montre les informations d’un client comme NOM.Prénom ces deux informations peuvent être redondante mais c’est impossible d’avoir pleusieurs client avec un seul numéro de téléphone et idem pour l’Email c’est-à-dire index unique

La table entre deux la relation entre ces deux tables



Le MCD complet sera fourni en annexe avec les autres documents

## Analyse des risques technique

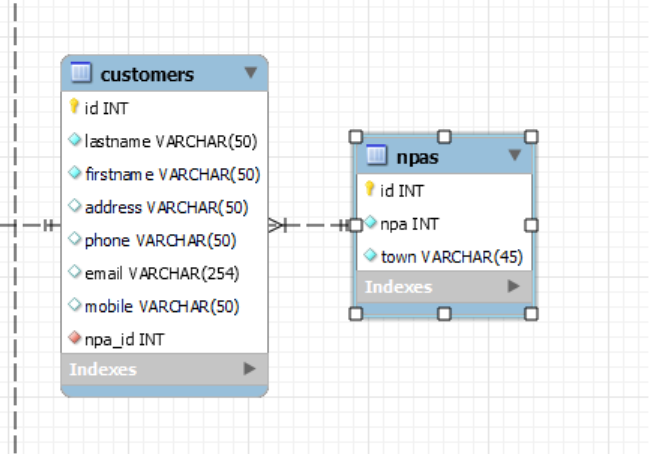
Les resique techniques en tent que développeur se produisent indépendant de moi

Un risque majour au niveau la santé un malalise et que cla prend beaucoup de temps

|  |  |  |
| --- | --- | --- |
| Risque Technique | Solution | Dégâts |
| Le laptop soit volé ou tombé en panne | Utiliser le poste de l’école  Et récupérer le code depuis github  Clone le repository  Git pull | Matériel  Et peut-être quelques modifications |
| Un incendie qui se produit soudainement | Faire  Git commit  Git push  et sortir | Matériel  Et peut-être quelques modifications |
| Un conflit entre les deux postes  Laptop et celui de l’école | On peut visualiser les fichiers qui ont était écraser | Au niveau totle de l’ensemble de la projet |
| Oublie le mot de passe de l’accès au GitHub et ses sites de gestion de projet | Récupérer mais cela prend de temps | Ralentir l’avancement de projet |
| Un problème est compliqué | De demande de l’aide au personne compétente | Ralentir l’avancement de projet |
| GitHub due à une mise à jour ou bien, un cyber attaque à leurs serveurs. | Stockage interne soit sur un disque dur ou bien sur un cloud | Blocage total pour récupérer les projet entier. |

## Modèle logique des données (MLD)

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements. Ainsi, le modèle logique est dépendant du type de base de données utilisé



Les deux tables contiennent sa propre clé première, la table de customer est liée à l’autre table par sa clé étrangère et il n’est pas vide il contient une valeur par défaut dépendent de l’autre table .

**Customers:**

**Column Name Champs**

* Id : une clé primaire et unique et par défaut il faut une valeur et incrémentation automatique
* lastname  : un champ avec une valeur par défaut
* firstname  : un champ avec une valeur par défaut
* address  :
* phone  :
* email  :
* mobile  : un champ avec une valeur par défaut et unique
* npa\_id  : un champ avec une valeur par défaut

**NPA :**

**Column Name champs**

* id : une clé primaire et unique et par défaut il faut une valeur et incrémentation automatique
* npa : un champ avec une valeur par défaut
* town : un champ avec une valeur par défaut

**INT**

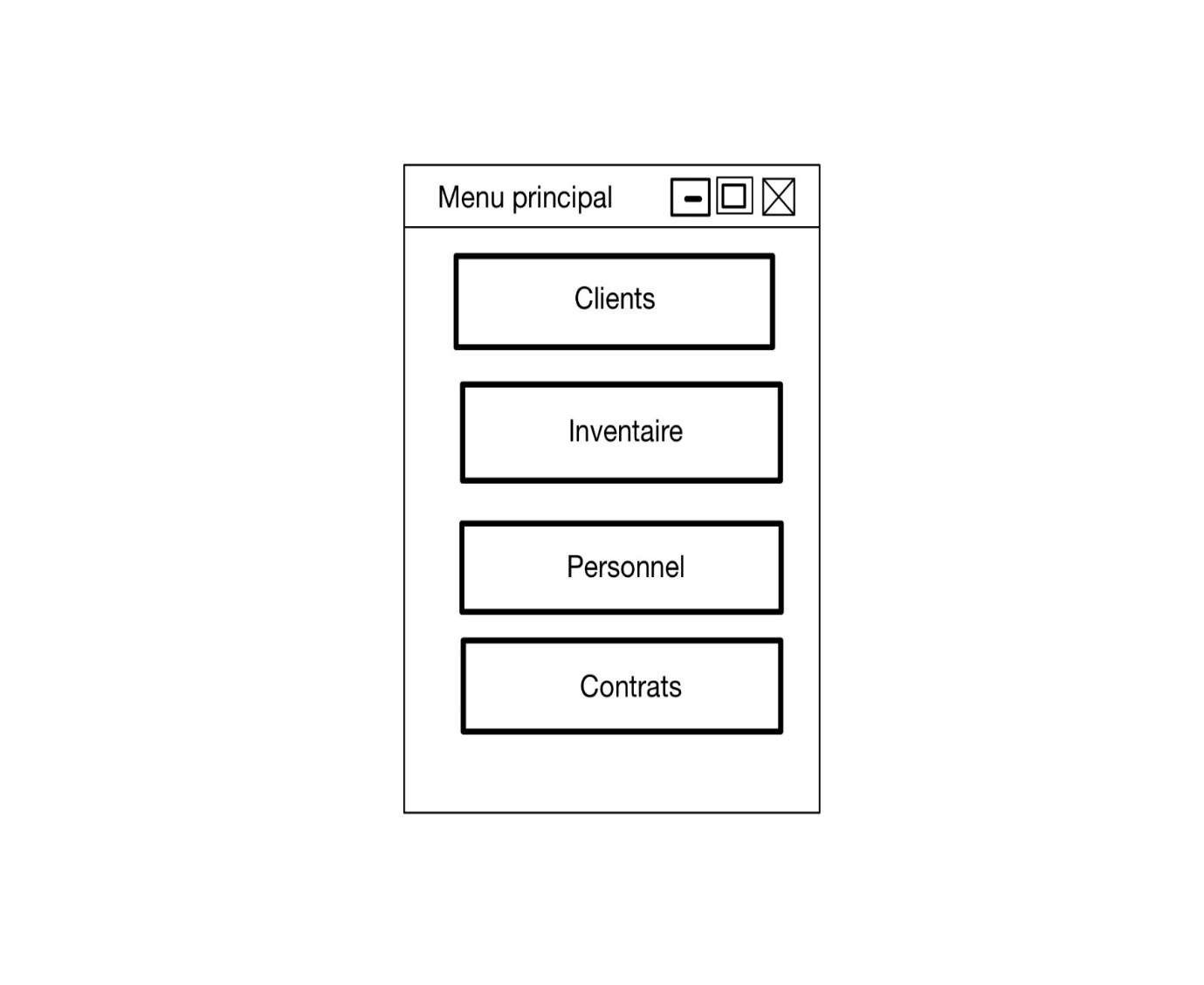
Est une abréviation qui peut signifier : integer (en français, entier), un type de données ; INT, en BASIC, une instruction qui permet de garder la partie entière d'un nombre

**VARCHAR(50)** : Données de type chaîne de taille 50 caractères

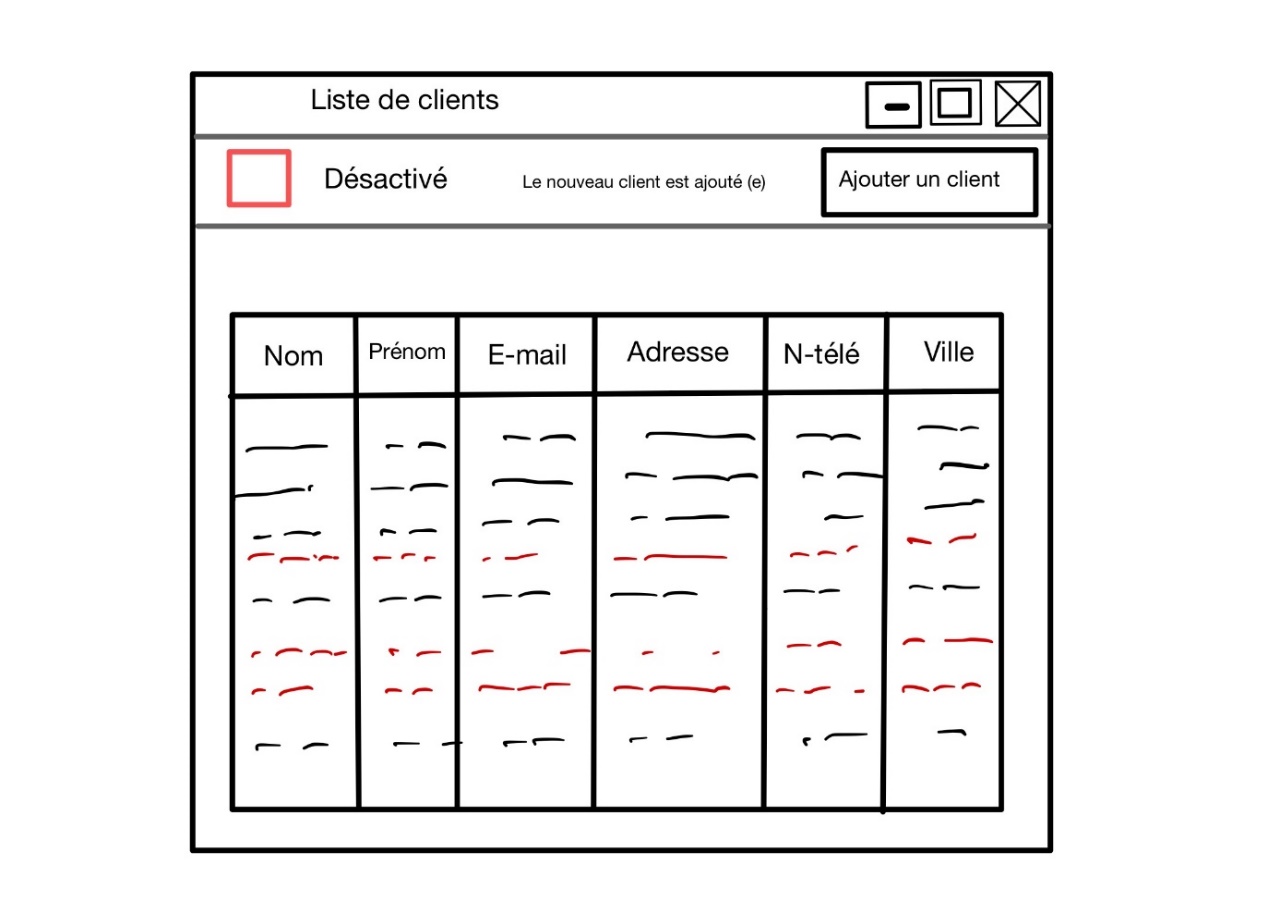
## Maquettes

J’ai utilisé une tablette IPad pro 2021 pour dessiner les maquettes et par une application Good Note.

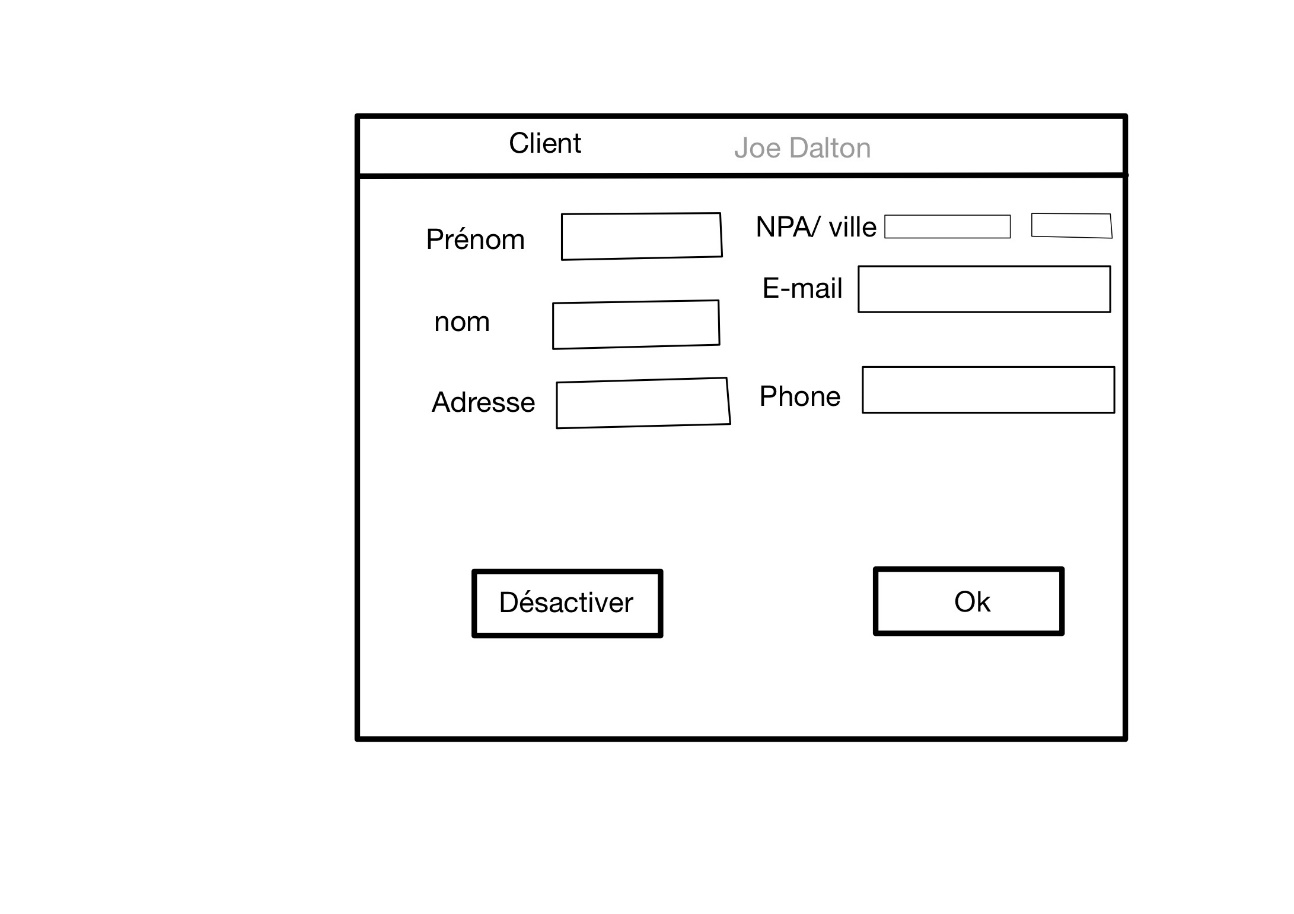
Menu au démarrage de SPLKS Quand on clique bouton Clients



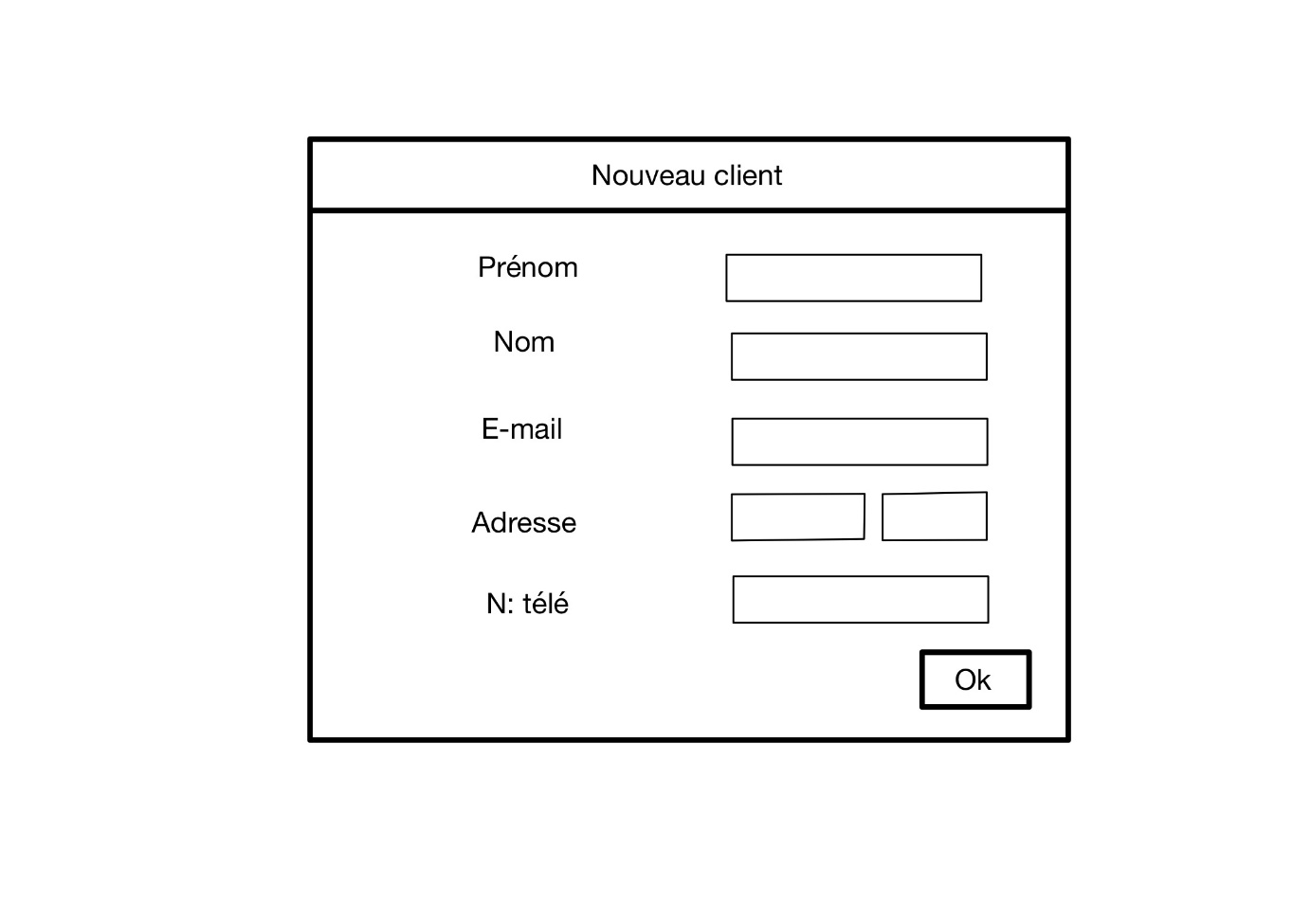
La liste de client s’affiche avec les clients

**

Quand je clique sur une case cette Ce formulaire s’affiche



Quand je clique sur ajouter un client Formulaire d’ajout s’affiche

**

Mes stories

**

## Stratégie de test

**Version développeur : dans un enivrement Windows os version 10**

Je vais installer la base de données sur un laptop et tenter faire la connexion via l’adresse IP et une fois la connexion établie je vais configure les paramètres sur un autre laptop et lancer Sploks et voir comme va fonctionner via cette méthode puis tenter d’en faire sur plusieurs laptops pour voir si la base de données est réceptive sur plusieurs demandes au même temps

**Version utilisateur : dans un enivrement Windows os version 10**

Le format final de l’application sera en **.exe** et puis la base de sonnées sera installé sur un serveur LAMP comme mentionné auparavant il est indépendant de tous les autres postes du magasin.

Solpks : sera installer sur tous les poste qui sont 5 + un serveur à ajouter pour la base de données

Pour les testeurs il y a un temps de transitions de Coliks à Sploks et comme c’est connu souvent le début est un difficile

## Dossier de conception

* Je travaille sur mon laptop privé ([X1 Yoga 3rd Gen)](https://www.lenovo.com/ch/en/laptops/thinkpad/thinkpad-x1/ThinkPad-X1-Yoga-3rd-Gen/p/22TP2TXX13Y?orgRef=https%253A%252F%252Fwww.google.com%252F) et j’ai le poste celui de l’école.
* Le système de réalisation c’est Windows 11 os et d’utilisation c’est Windows 10 os il y n’aura pas de problèmes au vu que Windows 11 est supérieur
* Les maquettes sont dessinées par [Goodnote](https://www.goodnotes.com/) via une tablette IPad pro 2021
* Réalisation les maquettes c’est via [QT Desinger](https://build-system.fman.io/qt-designer-download) : est l'outil Qt pour concevoir et créer des interfaces utilisateur graphiques (GUI) avec Qt Widgets. Vous pouvez composer et personnaliser vos fenêtres ou boîtes de dialogue de manière WYSIWYG (ce que vous voyez, c'est ce que vous obtenez) et les tester à l'aide de différents styles et résolutions. Donc les maquettes et les interfaces graphiques sur Sploks
* [MySQL](https://www.mysql.com/) Workbench : Un logiciel de gestion et d'administration de bases de données MySQL
* [PyCharm](https://www.jetbrains.com/pycharm/) est un environnement de développement intégré utilisé pour programmer en Python ou un IDE de votre choix. Comme Visual studio code.
* [Python](https://www.jetbrains.com/pycharm/) : est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions
* [Icescrum](https://icescrum.cpnv.ch/) est une plateforme pour gestion de projet et découpage le projet par sprint et chaque sprint représente une ou plusieurs stories et chaque story contient de tests.
* [IceScrub](https://github.com/XCarrel/IceScrub.git) une application développer par M.Carrel pour calculer le temps effectif sur chaque sprint , story, tâche, et le temps totale.

J’ai utilisé ces logiciels en version gratuite car c’est open source et bien sûr qu’il y a d’autre moyen comme d’autre logiciels.

L’app SPLOKS va fonctionner d’un enivrement Windows OS 10 et il sera lancer en format .exe

***CODE: SQL***

***1. Display All users***

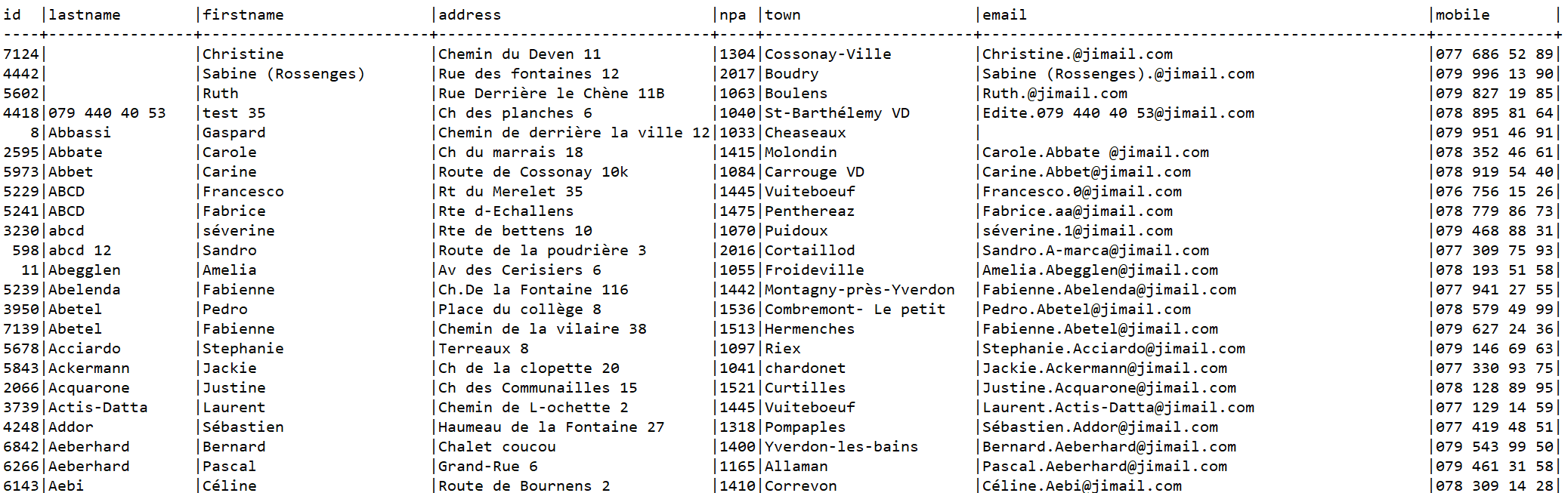
[Request]

*SELECT customers.id, lastname,firstname,address,npas.npa,npas.town,email,mobile*

*FROM customers*

*inner join npas on npas.id = customers.npa\_id* ***order by*** *lastname*

[Response]

**

***2. Display Details***

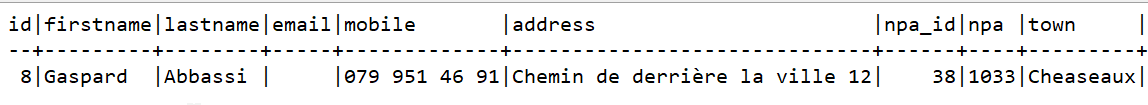
[Request]

*SELECT customers.id, firstname,lastname,email,mobile,address,npa\_id,npas.npa,npas.town FROM customers*

*inner join npas on npas.id = customers.npa\_id*

*Where customers.id = 8 Order by customers.id*

[Response]

**

***3. update one raw***

[Request]

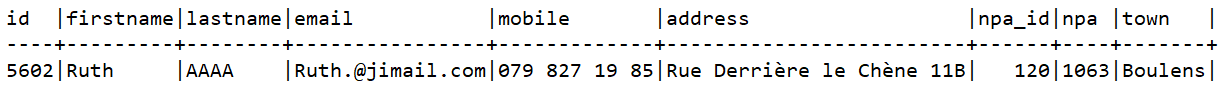
*UPDATE customers*

*Inner join npas on npas.id = customers.npa\_id*

*SET firstname='Ruth',lastname='AAAA',address='Rue Derrière le Chène 11B',email='Ruth.@jimail.com',mobile= '079 827 19 85', npa= 1063 ,town= 'Boulens'*

*WHERE customers.id = 5602*

[Response]



# Réalisation

## Dossier de réalisation

**Les versions des systèmes d'exploitation et des outils logiciels**

|  |  |
| --- | --- |
| Logiciel | Version |
| Windows | Windows 11 Pro,  Version 10.0.22000 |
| MySQL | 8.0.27 |
| Python | 3.10.2 |
| PIP | 22.1 |
| QtDesigner | 5.11.1 |

**Structure du repository**

Certains dossiers de Python moins pertinents ont été enlevés.

Seulement les dossiers et

Les fichiers à la racine sont affichés. Uniquement ceux que j'ai utilisé sont définis.

L’application installé sur mon laptop le chemin : C:\Users\marwa\Documents\GitHub\TPI-SPLOKS\

[**Sploks**](https://github.com/alhelo-93/TPI-SPLOKS/tree/main/Sploks)

**.**

**.** Const.py

**.** Sploks.py

**.**

**.** ──────────**controler**

**.** │ clientsControler.py

**.** │ contractsControler.py

**.** │ menuControler.py

**.** │ staffsControler.py

**.**

**.**

**.** ──────────**model**

**.** │ client.py

**.** │ contract.py

**.** │ helpers.py

**.** │ mold.py

**.** │ staff.py

**.**

**.**

**.** ─────────**views**

│ clients.ui

│ clientsDetails.ui

│ historylist.ui

│ menu.ui

│ newclient.ui

│ staffs.ui

│ staffsDetails.ui

**Liste de paquet**

Installer l’implémentation de code connexion BD et interface graphique

|  |  |
| --- | --- |
| Paquet | Version |
| **Setup Tools** | **60.2.0** |
| **pip** | **21.3.1** |
| **PyQt5-sip** | **12.10.1** |
| **Mysql-cnnector-python** | **8.0.29** |

**Programmation et scripts**

Créer un trigger (déclencheur)sur la table Customer pour écouter une mise à jour faite et déclencher l'exécution d'une instruction, ou d'un bloc d'instructions, lorsqu'une, ou plusieurs lignes sont insérées et l’enregistrer sur la table logs.

Transfer une valeur une fonction à une quatre en utilisant une variable globale par exemple pour afficher un message qu’un client est bien modifié sur la fenêtre de la liste de clients

Du formulaire d’édition envoyer une valeur du client enregistré à la fenêtre de client

CREATE DEFINER=`root`@`localhost` TRIGGER `customers\_AFTER\_UPDATE` AFTER UPDATE ON `customers` FOR EACH ROW BEGIN

SET @Change = '';

IF OLD.firstname <> new.firstname THEN

SET @Change = CONCAT ('Update Student Record ', OLD.firstname);

INSERT INTO logs(clientId,text) VALUES(old.id,@Change);

END IF;

IF OLD.lastname <> new.lastname THEN

SET @Change = CONCAT('Update Student Record ', OLD.lastname);

INSERT INTO logs(clientId,text) VALUES(old.id,@Change);

END IF;

IF OLD.address <> new.address THEN

SET @Change = CONCAT ('Update Student Record ', OLD.address);

INSERT INTO logs(clientId,text) VALUES(old.id,@Change);

END IF;

IF OLD.email <> new.email THEN

SET @Change = CONCAT ('Update Student Record ', OLD.email);

INSERT INTO logs(clientId,text) VALUES(old.id,@Change);

END IF;

IF OLD.mobile <> new.mobile THEN

SET @Change = CONCAT ('Update Student Record ', OLD.mobile);

INSERT INTO logs(clientId,text) VALUES(old.id,@Change);

END IF;

END

## Description des tests effectués

1. **Afficher la liste de clients**

|  |  |
| --- | --- |
| Identifiant |  |
| En tant que | Utilisateur |
| Je veux | Afficher la liste de clients |
| Pour | Consulter les informations d’un client |
| Priorité | M |

|  |  |  |
| --- | --- | --- |
| Action | Condition | Réaction |
| Appuyer sur clients |  | Une fenêtre s’affiche avec la liste de clients |
| Appuyer sur X |  | L’application se ferme |
|  |  |  |

1. **Afficher le formulaire d’édition**

|  |  |
| --- | --- |
| Identifiant |  |
| En tant que | Utilisateur |
| Je veux | Afficher la détaille d’un client |
| Pour | Pour modifier |
| Priorité | M |
|  |  |

|  |  |  |
| --- | --- | --- |
| Action | Condition | Réaction |
| Appuyer sur clients |  | Une fenêtre s’affiche avec la liste de clients |
| Appuyer sur le linge dans la liste |  | Un formulaire d’édition s’affiche |
| Je modifie les champs |  | Les informations d’un client changent |
| Je clique ok |  | Les informations modifiées s’enregistrent |
|  |  | Le formulaire se ferme |
|  |  | La liste de client se mets à jour |
| Je clique annuler |  | Le formulaire se ferme |

1. **Afficher le formulaire d’ajout**

|  |  |
| --- | --- |
| Identifiant |  |
| En tant que | Utilisateur |
| Je veux | Afficher le formulaire d’ajout d’un client |
| Pour | Pour ajouter |
| Priorité | M |
|  |  |

|  |  |  |
| --- | --- | --- |
| Action | Condition | Réaction |
| Appuyer sur clients |  | Une fenêtre s’affiche avec la liste de clients |
| Appuyer sur le bouton ajouter un client |  | Un formulaire d’ajout d’un client s’affiche |
| J’insère les champs |  | Les informations d’un client restent dans son champ |
| Je clique ok |  | Les informations s’enregistrent |
|  |  | Le formulaire se ferme |
|  |  | La liste de client se mets à jour |
| Je clique annuler |  | Le formulaire se ferme |

1. **Chercher un client par son numéro de téléphone ou son email**

|  |  |
| --- | --- |
| Identifiant |  |
| En tant que | Utilisateur |
| Je veux | Chercher un client |
| Pour | Pour consulter |
| Priorité | M |
|  |  |

|  |  |  |
| --- | --- | --- |
| Action | Condition | Réaction |
| Appuyer sur clients |  | Une fenêtre s’affiche avec la liste de clients |
| Dans le bar de recherche je taper le numéro de téléphone ou son email |  | Le propriétaire de ce numéro s’affiche |

***Test effectués***

|  |  |  |
| --- | --- | --- |
| *Scénario* | *31 mai 2022 développeur MAO* | Attente de Monsieur Carrel pour les tests finaux |
| **Afficher la liste de clients** | *Ok* |  |
| **Afficher le formulaire d’édition** | *Ok* |  |
| **Afficher le formulaire d’ajout** | *OK* |  |
| **Chercher un client par son numéro de téléphone**  **Ou son email** | *OK* |  |

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

* Alhelo-Sploks-rapport
* Journal de travail
* Timesheet
* Readme : contient la documentation d’installation du projet en tant que développeur

# Conclusions

***Objectifs atteints***

|  |  |
| --- | --- |
| Objectif | Atteint ? |
| * Saisir les informations d'un nouveau client | OK |
| * Modifier les informations d'un client existant | OK |
| * Parcourir le carnet d'adresse | OK |
| * Désactiver un client | - |
| * Gérer des liens entre clients | - |
| * Consulter l'historique | OK |
| * Enregistrement des actions | - |

***Difficultés :***

Au début de sprint j’ai rencontré une série de problèmes et mon application est complétement bloqué et j’ai passé le temps à fix les bugs au début les erreurs qui s’affichent semblant simple mais c’est au contraire car sur le site stackoverflew montre que c’est genre de problème connu avec os windows 11 et ce qui a basculé ma palification initiale

***Points positifs :***

Plan personnel j’ai développé une stratégie à débugger les bugs malgré le temps perdu mais fixer un bug quelquefois c’est plus compliqué que prévu et surtout si on utilise le MVC.  
je suis content de développer ce point sur le Bilan personnel et professionnel

**Suites possibles :**

Il sera indépendant de moi et selon les besoins du client et leur feedback,

Carrément oui il y a une suite.

# Annexes

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

## Sources – Bibliographie

|  |  |
| --- | --- |
| *Site* |  |
| [stackoverflow](https://stackoverflow.com/) | *Pour trouver une solution pour les bugs* |
| [*MySQL Tutorial*](https://www.mysqltutorial.org/) | *Consulter pour faire mes requêtes* |

## Journal de travail

Il sera fourni en document séparé de ce rapport

## 

## Manuel d'Installation

* En tant qu’un développeur : sur le site GitHub sur mon repository de ce projet readme
* Entan qu’un utilisateur :
* Il faut installer le nouveau serveur
* Puis configuration du réseau l’adresse IP
* Installer sploks en format .exe

## Manuel d'Utilisation

Il sera fourni séparé de ce rapport

## Archives du projet

*Media, … dans une fourre en plastique*