



# SPLOKS

Projet TPI 4<sup>ème</sup> année

## Résumé

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

<p>Auteur : Marwan Alhelo E-mail : <a href="mailto:marwan.alhlo@cpnv.ch">marwan.alhlo@cpnv.ch</a> Classe : SI-C4a Date : 24.05.2022</p>
---

Table des matières

1	Analyse préliminaire .....	2
1.1	Introduction .....	2
1.2	Objectifs.....	3
1.3	Planification initiale .....	4
2	Analyse / Conception.....	5
2.1	Concept .....	5
2.2	Stratégie de test.....	<b>Erreur ! Signet non défini.</b>
2.3	Risques techniques .....	<b>Erreur ! Signet non défini.</b>
2.4	Planification .....	15
2.5	Dossier de conception .....	<b>Erreur ! Signet non défini.</b>
3	Réalisation.....	15
3.1	Dossier de réalisation .....	15
3.2	Description des tests effectués .....	16
3.3	Erreurs restantes .....	16
3.4	Liste des documents fournis .....	16
4	Conclusions .....	16
5	Annexes.....	18
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	18
5.2	Sources – Bibliographie .....	18
5.3	Journal de travail .....	18
5.4	Manuel d'Installation .....	18
5.5	Manuel d'Utilisation.....	18
5.6	Archives du projet.....	18

# 1 Analyse préliminaire

## 1.1 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.

## 2 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,

### 2.1.1 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

### 2.1.2 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).

### 2.1.3 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.

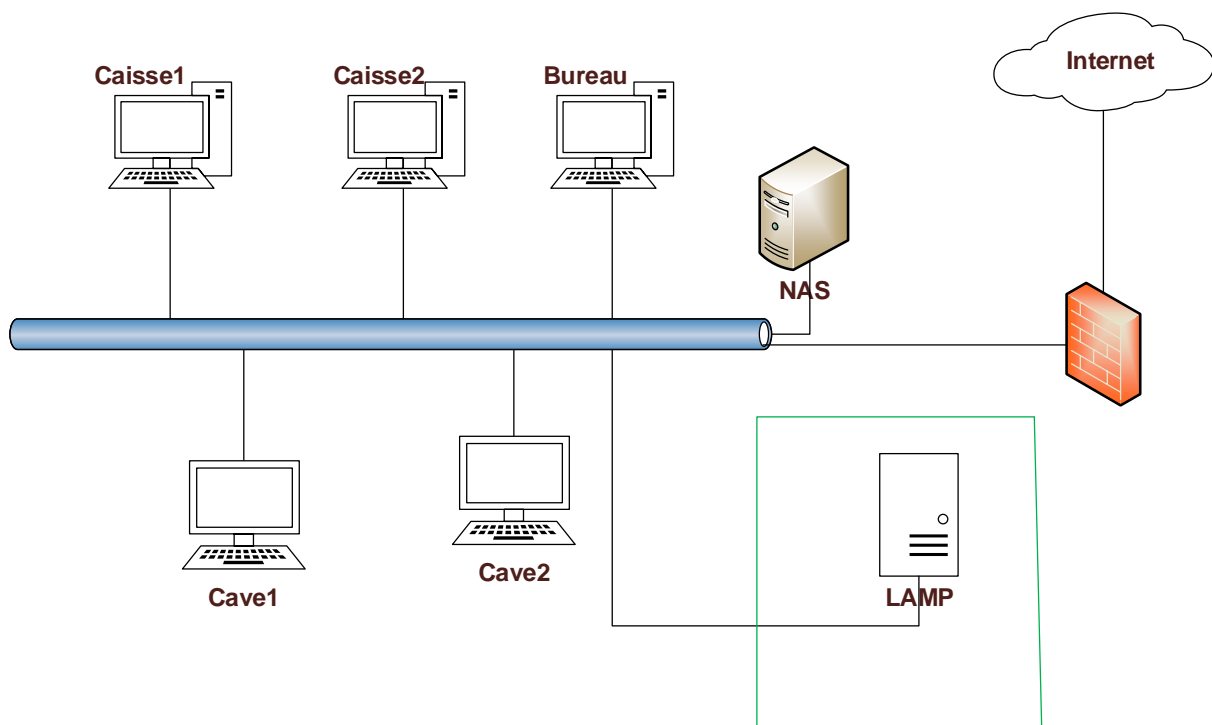
## 3 Planification initiale

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	
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%	

## 4 Analyse / Conception

### 4.1 Concept

- Le schéma du réseau au magasin de sports et Sploks va fonctionner dans cet environnement
- 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 données
- Les postes ce sont de ordinateurs avec windows 10 comme system D'exploitation



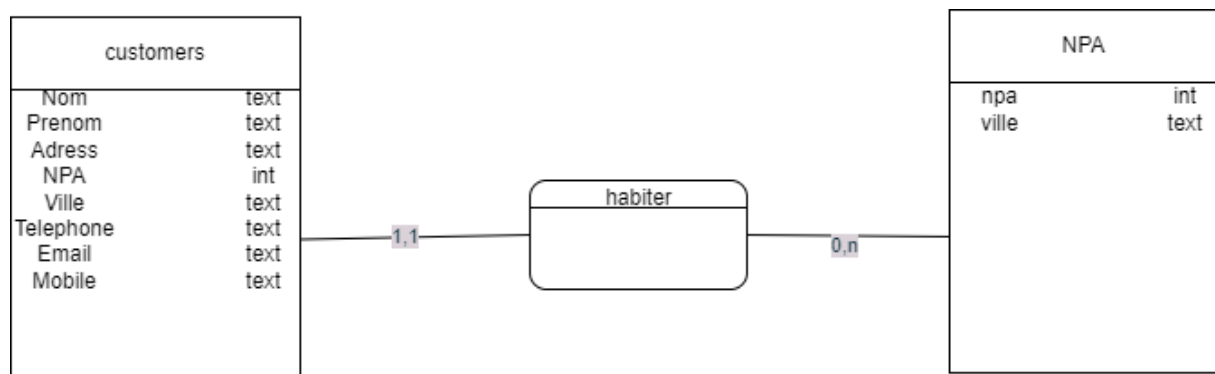
#### 4.2 Modèle conceptuel de données (MCD)

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 plusieurs client avec un seul numéro de téléphone et idem pour l'Email

La table entre deux la relation entre ces deux tables



Le MCD complet sera fourni en annexe avec les autres documents

### 4.3 Analyse des risques technique

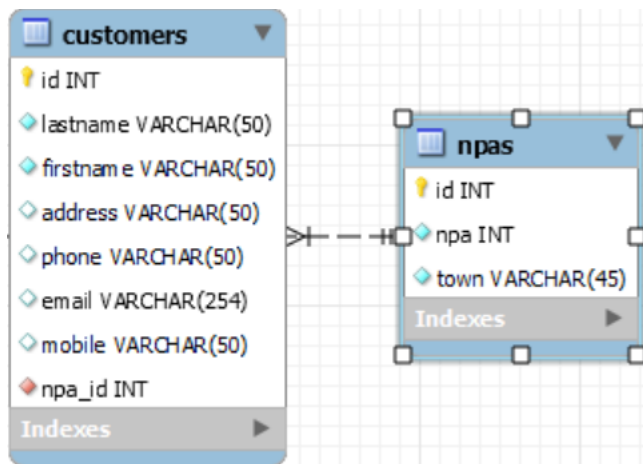
Les risques techniques en tant que développeur se produisent indépendamment de moi  
Un risque majeur au niveau de la santé, une maladie et que cela prend beaucoup de temps

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 été écrasés	Au niveau total 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	Demande de l'aide à une 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 le projet entier.



#### 4.4 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 dépendant 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

## 4.5 Maquettes

J'ai utilisé une tablette Ipad pro 2021 pour dessiner les maquette et par une application Good Note.

## Menu au démarrage de SPLKS

### Quand on clique bouton Clients

Menu principal

Clients

Inventaire

Personnel

Contrats

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

[illegible]

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

Client		Joe Dalton	
Prénom	<input type="text"/>	NPA/ ville	<input type="text"/>
nom	<input type="text"/>	E-mail	<input type="text"/>
Adresse	<input type="text"/>	Phone	<input type="text"/>
<input type="button" value="Désactiver"/>		<input type="button" value="Ok"/>	

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

Nouveau client	
Prénom	<input type="text"/>
Nom	<input type="text"/>
E-mail	<input type="text"/>
Adresse	<input type="text"/> <input type="text"/>
N: télé	<input type="text"/>
<input type="button" value="Ok"/>	



#### 4.6 Stratégie de test

##### Version développeur : dans un environnement 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 environnement Windows os version 10

Le format final de l'application sera en **.exe** et puis la base de données 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

- 
- Décrire la stratégie globale de test:
- 
- types de des tests et ordre dans lequel ils seront effectués.
- les moyens à mettre en œuvre.
- couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).
- données de test à prévoir (données réelles ?).
- les testeurs extérieurs éventuels.

#### 4.7 Dossier de conception

- Je travaille sur mon laptop privé ([X1 Yoga 3rd Gen](#)) 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](#) via une tablette iPad pro 2021
- Réalisation les maquettes c'est via [QT Desinger](#) : 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](#) Workbench : Un logiciel de gestion et d'administration de bases de données MySQL
- [PyCharm](#) est un environnement de développement intégré utilisé pour programmer en Python ou un IDE de votre choix. Comme Visual studio code.
- [Python](#) : 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](#) 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](#) une application développer par M.Carrel pour calculer le temps effectif sur chaque sprint , story, tâche, et le temps totale.

- Le choix des outils logiciels pour la réalisation et l'utilisation
- Site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...
- Bases de données : décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.  
Programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...

## 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]

- 
- | id |lastname |firstname | address |npa |town |email| mobile|
- |-----|-----|-----|-----|-----|-----|-----|
- | | | | | | | | |

- 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]
- | id |lastname |firstname | address |npa\_id|npa |town |email| mobile|
- |-----|-----|-----|-----|-----|-----|-----|
- | | | | | | | | |
- |

- 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 Chêne`
- `11B',email='Ruth. @jimail.com',mobile='079 827 19 85', npa= 1063 ,town=`
- `'Boulens'`
- `WHERE customers.id = 5602`

## 4.8 Planification

*Révision de la planification initiale du projet :*

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

## 5 Réalisation

### 5.1 Dossier de réalisation

*Décrire la réalisation "physique" de votre projet*

- *les répertoires où le logiciel est installé*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*
- *le numéro de version de votre produit !*
- *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Évitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...*



## 5.2 Description des tests effectués

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 de la client enregistré à la fenêtre de client

*Pour chaque partie testée de votre projet, il faut décrire:*

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)
- tests sans preuve: fournir au moins une description

## 5.3 Erreurs restantes

*S'il reste encore des erreurs:*

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

## 5.4 Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

- le rapport de projet
- le manuel d'Installation (en annexe)
- le manuel d'Utilisation avec des exemples graphiques (en annexe)
- autres...

## 6 Conclusions

*Développez en tous cas les points suivants:*

- Objectifs atteints / non-atteints

- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

## 7 Annexes

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

### 7.2 Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)*

### 7.3 Journal de travail

Date	Durée	Activité	Remarques

7.4

### 7.5 Manuel d'Installation

### 7.6 Manuel d'Utilisation

### 7.7 Archives du projet

*Media, ... dans une fourre en plastique*