Résumé

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 :17.05.2022

Table des matières

[1 Analyse préliminaire 2](#_Toc499021832)

[1.1 Introduction 2](#_Toc499021833)

[1.2 Objectifs 3](#_Toc499021834)

[1.3 Planification initiale 4](#_Toc499021835)

[2 Analyse / Conception 5](#_Toc499021836)

[2.1 Concept 5](#_Toc499021837)

[2.2 Stratégie de test 9](#_Toc499021838)

[2.3 Risques techniques 9](#_Toc499021839)

[2.4 Planification 9](#_Toc499021840)

[2.5 Dossier de conception 9](#_Toc499021841)

[3 Réalisation 10](#_Toc499021842)

[3.1 Dossier de réalisation 10](#_Toc499021843)

[3.2 Description des tests effectués 10](#_Toc499021844)

[3.3 Erreurs restantes 10](#_Toc499021845)

[3.4 Liste des documents fournis 11](#_Toc499021846)

[4 Conclusions 11](#_Toc499021847)

[5 Annexes 12](#_Toc499021848)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 12](#_Toc499021849)

[5.2 Sources – Bibliographie 12](#_Toc499021850)

[5.3 Journal de travail 12](#_Toc499021851)

[5.4 Manuel d'Installation 12](#_Toc499021852)

[5.5 Manuel d'Utilisation 12](#_Toc499021853)

[5.6 Archives du projet 12](#_Toc499021854)

# 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’affaire (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



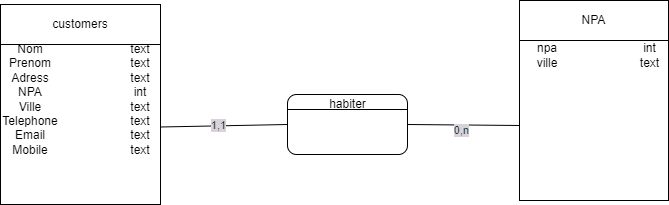
## Modèle conceptuel de données (MCD)

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

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 c’est une malaise au niveau de la santé

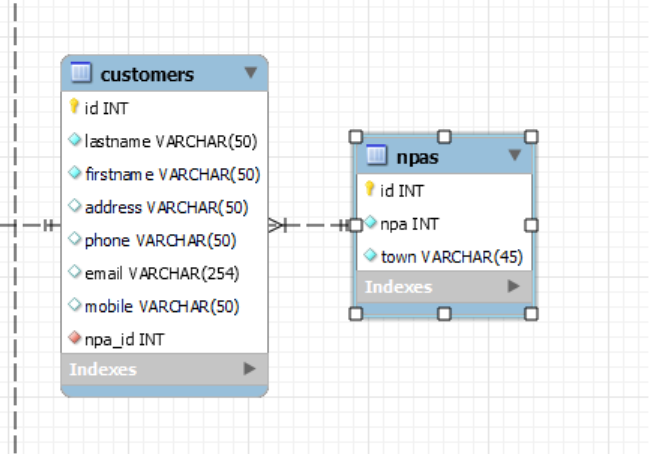
Car on utilse github bien connu pour le monde de programation et les risque sont quasiment faible car c’est saver d’un Claud qulque part

Personnellement j’utilise mon laptop privé et pour cela si mon laptop tombe en panne ou perdu il me manquera qu’une partie de code et de ce fait c’est rattrapable



## 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.

**Customers :**

**Column Name DataType NOTNULL primary Key Auto Increment**

* id INT x x
* lastname  : VARCHAR(50) x
* firstname  : VARCHAR(50) x
* address  : VARCHAR(50)
* phone  : VARCHAR(50)
* email  : VARCHAR(50)
* mobile  : VARCHAR(50)
* npa\_id  : INT

**NPA :**

**Column Name DataType NOTNULL primary Key Auto Increment**

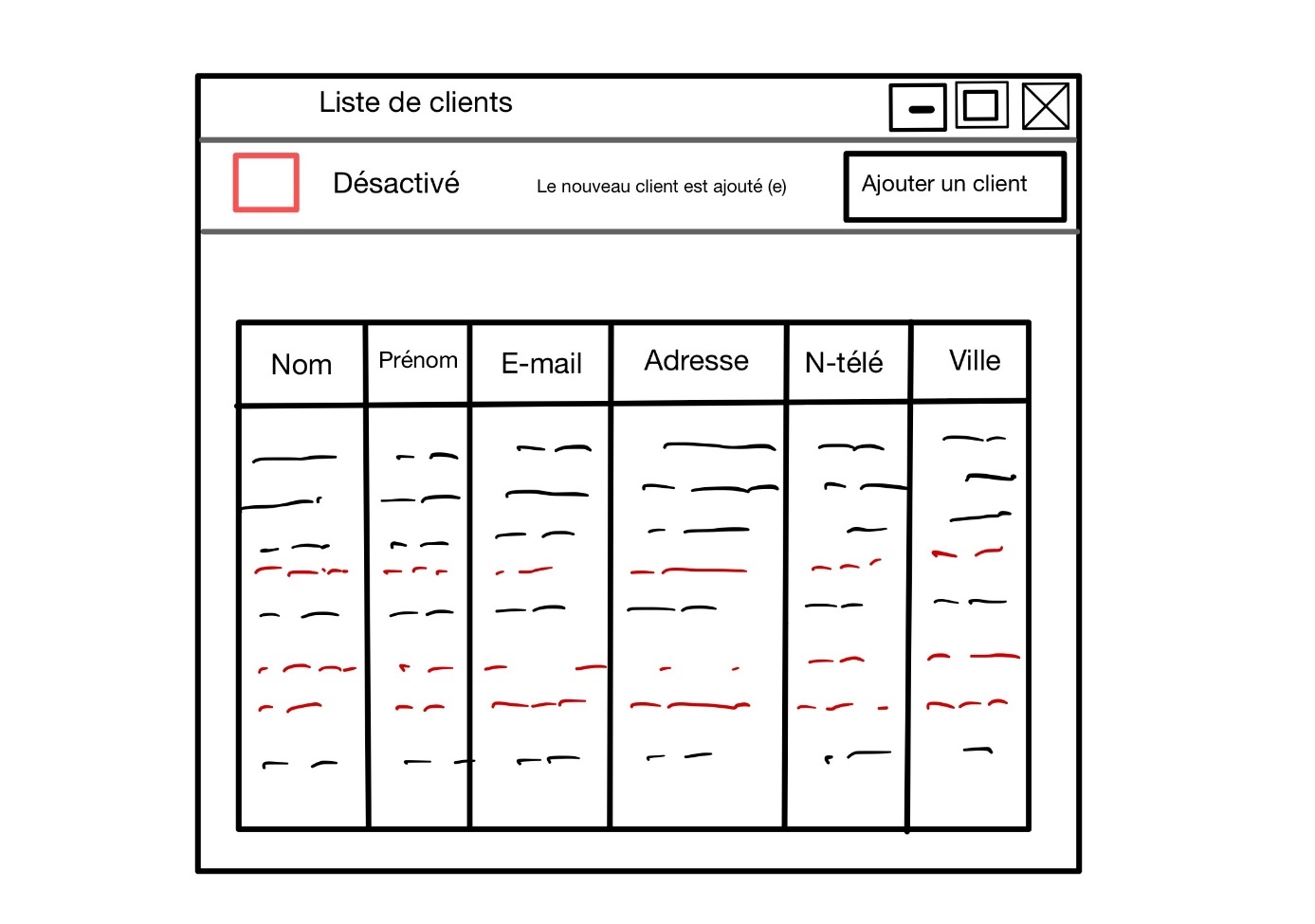
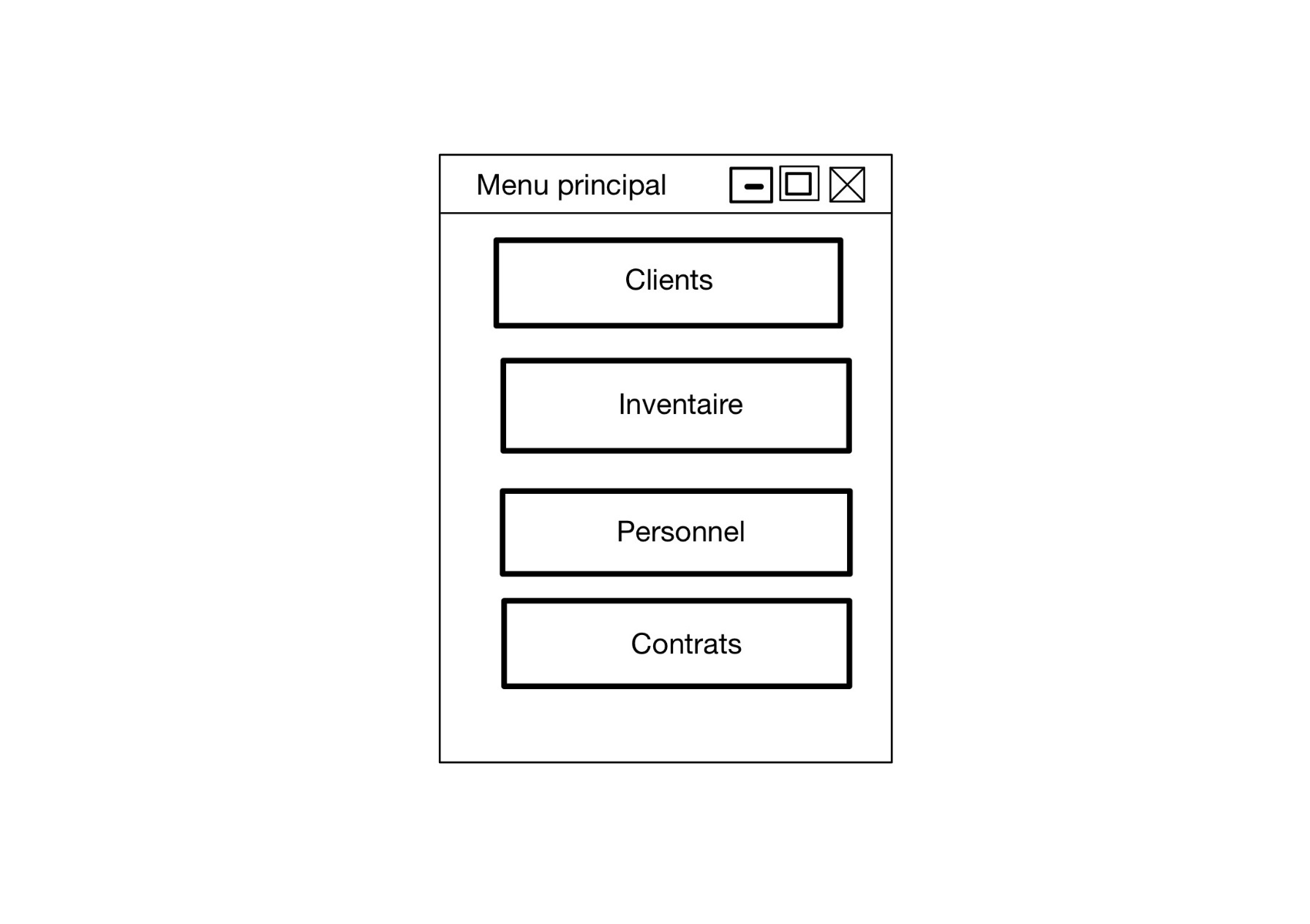
* id : INT x x
* npa : VARCHAR(50) x
* town : VARCHAR(50) x

**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

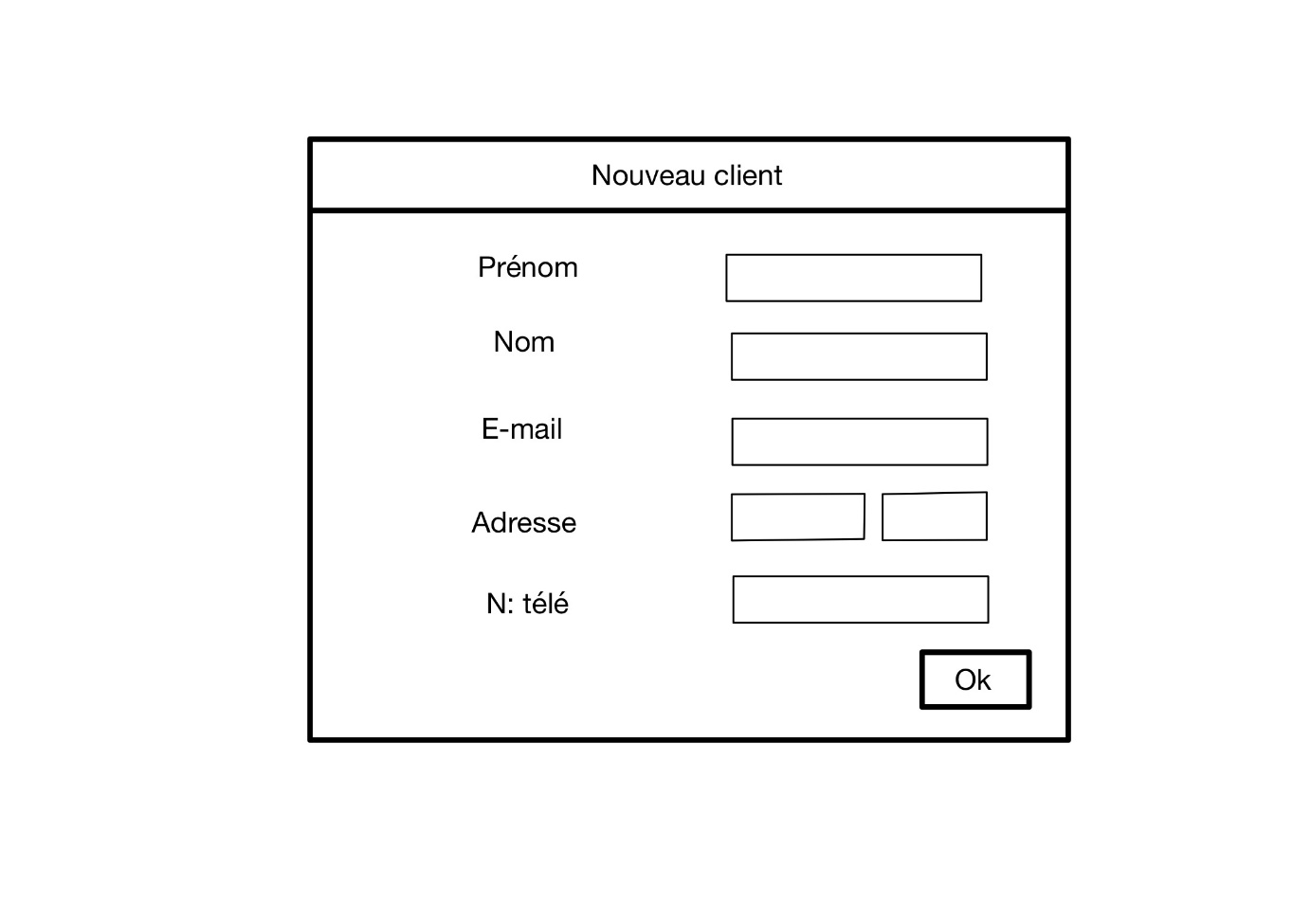
## Maquettes

**

Menu au démarrage de SPLKS

Quand on clique bouton Clients Quand je clique sur ajouter un client

Formulaire d’ajout s’affiche

**

*Le concept complet avec toutes ses* *annexes :*

*Par exemple :*

* *Multimédia: carte de site, maquettes papier, story board préliminaire, …*
* *Bases de données: interfaces graphiques, modèle conceptuel.*
* *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle……*

## Stratégie de test

* 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.

## 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 via une tablette IPad
* 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 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 ;
* 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…

## 

## Risques techniques

* *risques techniques (complexité, manque de compétences, …).*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, …).*

## 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 !***

# Réalisation

## 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 : Evitez 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…*

## Description des tests effectués

*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*

## Erreurs restantes

*S'il reste encore des erreurs:*

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

## 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…*

# Conclusions

*Développez en tous cas les points suivants:*

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

# Annexes

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

## 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)*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## 

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*