

# SPLOKS

---



Luís Pedro  
Fernandes Pinheiro

SI-C4a  
Projet TPI 2022

---

Table des matières

1	Analyse préliminaire .....	3
1.1	Introduction .....	3
1.2	Objectifs .....	4
1.2.1	Sprint 1 .....	4
1.2.2	Sprint 2 .....	4
1.2.3	Sprint 3 .....	4
1.2.4	Sprint 4 .....	4
1.3	Planification initiale .....	5
2	Analyse / Conception .....	6
2.1	Concept .....	6
2.1.1	Coliks .....	6
2.1.2	Schéma Réseau .....	7
2.1.3	Modèle conceptuel des données .....	8
2.1.4	Modèle logique des données .....	9
2.2	Stratégie de test .....	10
2.3	Risques techniques .....	10
2.3.1	GitHub .....	10
2.3.2	Problème avec disque de stockage .....	10
3	Réalisation .....	11
3.1	Erreurs restantes .....	11
3.1.1	Trop d'items à la génération d'un contrat PDF .....	11
3.2	Liste des documents fournis .....	11
3.2.1	Journal de travail .....	11
3.2.2	Journal de bord .....	11
3.2.3	Documentation de projet .....	11
3.2.4	Résumé du rapport .....	11
4	Conclusions .....	12
4.1	Objectifs atteints .....	12
4.2	Objectifs non-atteints .....	13
4.3	Points positifs / négatifs .....	13
4.3.1	Positifs .....	13
4.3.2	Négatifs .....	13
4.4	Suites possibles pour le projet .....	13
5	Annexes .....	14
5.1	Sources – Bibliographie .....	14
5.1.1	Doutes MySQL .....	14
5.1.2	Tutoriels Python .....	14
5.1.3	Utilisation de Reportlab .....	14
5.2	Journal de travail .....	14
5.3	Manuel d'Installation .....	14
5.4	Manuel d'utilisation .....	15

## 1 Analyse préliminaire

### 1.1 Introduction

Sploks est un 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 ce projet parce que je porte un grand intérêt par les applications développées en Python, c'est aussi un projet que je connais déjà car il a été commencé un camarade de classe et moi pendant notre module de projet à deux. Ce projet est aussi très intéressant car il y a un possible vrai client qui pourra utiliser l'application final, Sports-Time d'Echallens.

## 1.2 Objectifs

Nous objectifs seront séparés par sprint, chaque sprint contiendra deux, voire 3 objectifs atteignables.

Ils sont planifiés avant le début du projet et inclus dans la planification initiale à côté de leur sprint correspondant.

### 1.2.1 Sprint 1

- Planifier la totalité du projet. Préparer les futurs sprints avec des objectifs S.M.A.R.T (Spécifique, Mesurable, Ambitieux, Réaliste, Temporel). Pour cela, il faut :
  - Accorder les modalités de travail avec le client
  - Créer la première version de la documentation de projet
  - Préparer l'environnement de travail dans les machines où je vais travailler pour ne pas perdre de temps plus tard.
- Débuter le développement de la première fonctionnalité, Établir un contrat de location :
  - Création de l'affichage d'un contrat
  - Afficher le contenu de la base de donnée dans celle-ci
  - Utiliser que le clavier pour naviguer

### 1.2.2 Sprint 2

- Établir un contrat de location, finalisation de la fonctionnalité avec les dernières demandes :
  - Pouvoir créer un utilisateur s'il n'existe pas
  - Imprimer 2 contrats à la fin de la création de celui-là
  - Pouvoir changer le prix final du contrat
- Consulter les contrats de location existants :
  - Pouvoir consulter les contrats d'un client
  - Afficher les détails
  - Possibilité d'imprimer le contrat

### 1.2.3 Sprint 3

- Gérer l'état des contrats de location
  - Pouvoir clôturer le contrat
  - Retour partiel du matériel
  - Lister les retours en retard

### 1.2.4 Sprint 4

- Finaliser le projet
  - Dernières corrections du code, corriger quelques fonctionnalités
  - Finaliser la documentation de projet et autres documents qui vont avec.

### 1.3 Planification initiale

La première partie du projet est à rendre pour le 31 mai 2022, ce sera la fin du TPI et début de la défense du projet.

Pour la gestion de projet de SPLOKS, on va utiliser l'outil IceScrum. IceScrum est un outil se base sur la méthodologie Agile et qui est majoritairement OpenSource (GPL/AGPL). Durant notre projet, on va aussi travailler avec l'outil GitHub qui lui-même utilise git pour faire du contrôle de version de nos fichiers. Cela permettra de travailler plus facilement et avancer sans peur de tout casser car on pourra toujours revenir à version précédente.

Sur GitHub, on utilisera aussi l'outil workflow git qui nous permet de séparer notre projet en plusieurs « branches », cela permet de travailler d'une façon plus sécurisée et de développer des nouvelles fonctionnalités sans que cela impact le travail fait précédemment.

Une branche appelée « main » contiendra le principal du projet, les versions fonctionnelles et être prêts à être utilisés se situeront là. Une deuxième branche appelée « develop » contiendra le contenu de « main » plus ce qui est fonctionnel mais pas validé par le Product Owner. On crée une nouvelle branche pour chaque nouvelle fonctionnalité qu'on commence à développer et quand elle sera fonctionnelle et prête à être vérifiée par le Product Owner, on la merge dans « develop ».

Sprint	Dates	Jours de travail	Objectifs
1	02.05 – 06.05	4	Planification du projet : <ul style="list-style-type: none"><li>• Modalités de travail</li><li>• Création de la documentation</li><li>• Préparation de l'environnement de travail</li></ul> Développement de fonctionnalité : <ul style="list-style-type: none"><li>• Établir un contrat de location</li></ul>
2	09.05 – 13.05	4	Terminer développement de fonctionnalité : <ul style="list-style-type: none"><li>• Établir un contrat de location</li></ul> Développement de fonctionnalité : <ul style="list-style-type: none"><li>• Consulter les contrats de location existants</li></ul>
3	16.05 – 20.05	4	Développement de fonctionnalité : <ul style="list-style-type: none"><li>• Gérer l'état des contrats de location</li></ul>
4	23.05 – 31.05	5	Finaliser le développement : <ul style="list-style-type: none"><li>• Dernières corrections du code</li></ul> Documentation finale : <ul style="list-style-type: none"><li>• Propreté du code (commentaires, etc...)</li><li>• Finaliser la documentation (visuelle, conclusion)</li></ul>
Fin de Projet			

## 2 Analyse / Conception

### 2.1 Concept

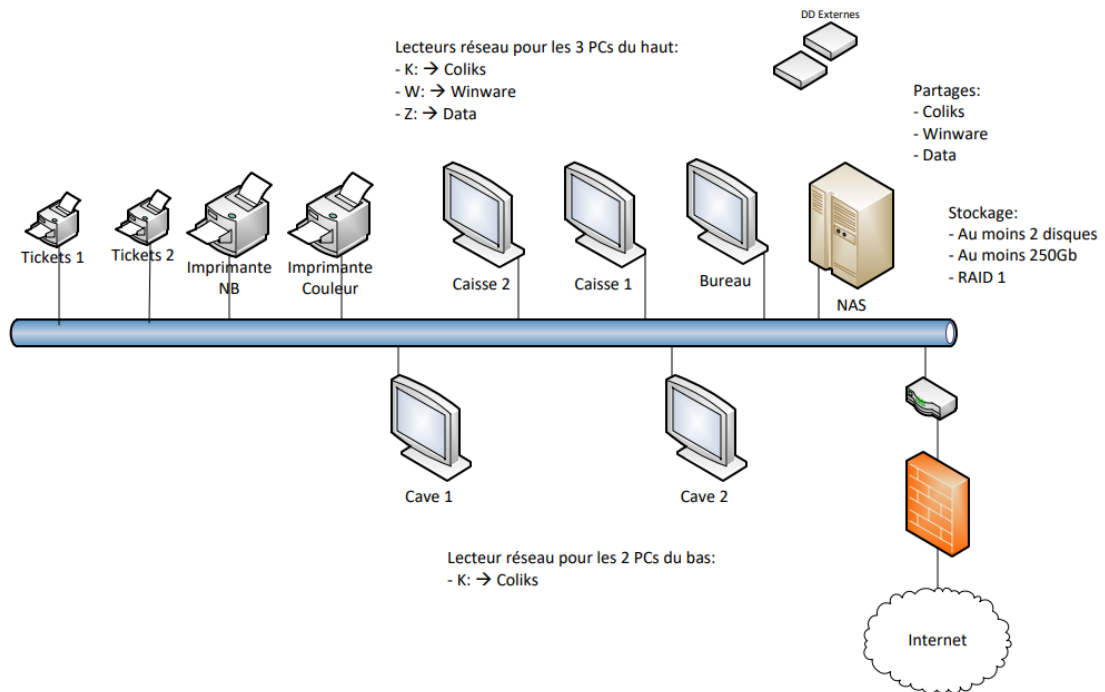
#### 2.1.1 Coliks

Coliks, une application en MSAccess, est la version actuelle utilisée par le magasin. Elle a été créée il y a environ 17 ans par Xavier Carrel. Ceci a été ma plus grande source d'inspiration, les interfaces de SPLOKS, ainsi que son fonctionnement ressemblent énormément à Coliks.

The image shows two parts of a software interface. On the left is a window titled 'Locations' with a 'Contrat' (Contract) form. The form includes search buttons 'Chercher Contrat Par Numéro' and 'Chercher Par Num Série'. It has input fields for 'Nom', 'Prénom', 'Adresse', 'Localité', 'Tél (fixe)', 'Natel', and 'Email'. Below these is a table with columns 'Pos', 'Objet', 'Cat', and 'Prix'. At the bottom of the form are fields for 'A retourner le', 'Total:', 'Pris le', 'Payé le', a checkbox for 'Rabais fin de saison (20%)', a 'Servi par' dropdown, and a 'Réglage par' dropdown. On the right is a sidebar menu for 'Coliks 2.7.4' with buttons for 'Inventaire', 'Clients', 'Prix de location', 'Formules', 'Retours en retard', 'Statistiques', 'Restockage', and 'Personnel'.

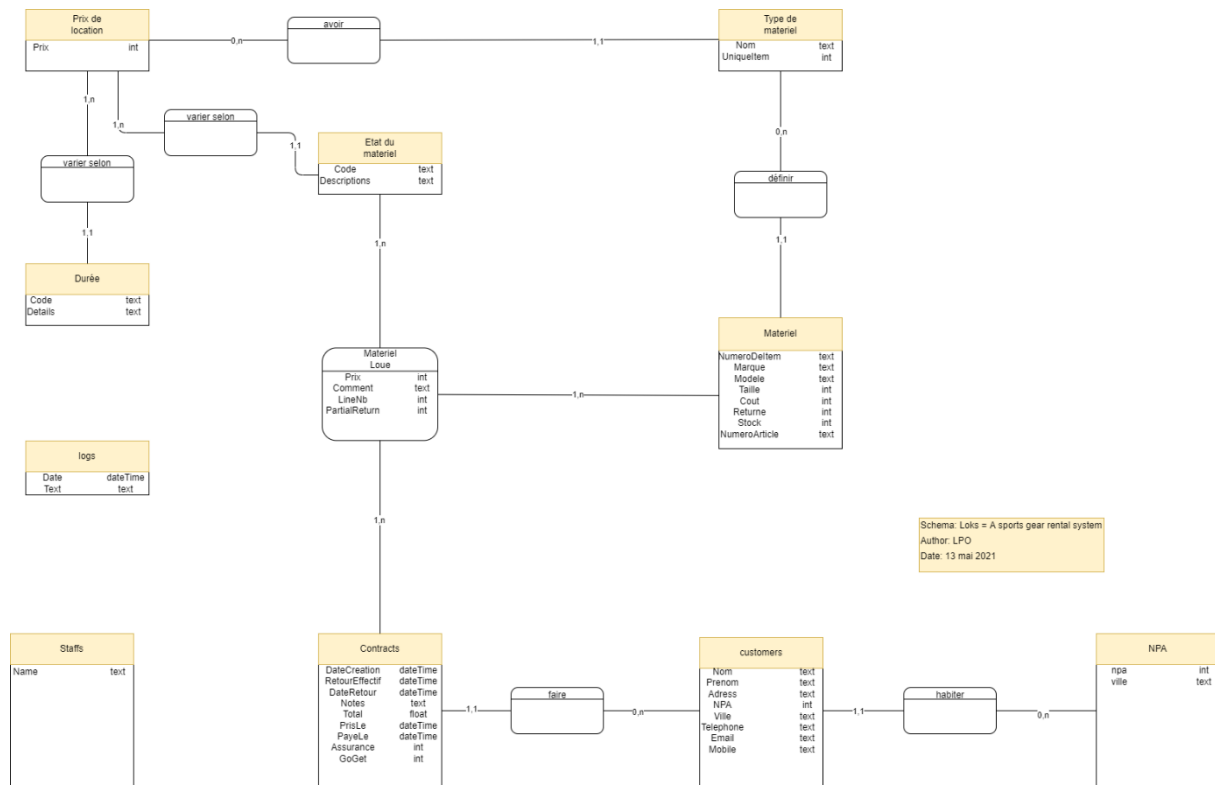
### 2.1.2 Schéma Réseau

Ceci est le schéma du réseau du magasin Sports-Time d'Echallens et ce sera sur cette environnement que SPLOKS devra fonctionner.



### 2.1.3 Modèle conceptuel des données

Le modèle conceptuel des données, MCD, a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Ceci a été créé au début du projet selon les besoins du client. Il nous permette facilement de distinguer les différentes tables de la base de données, ainsi que leurs attributs.



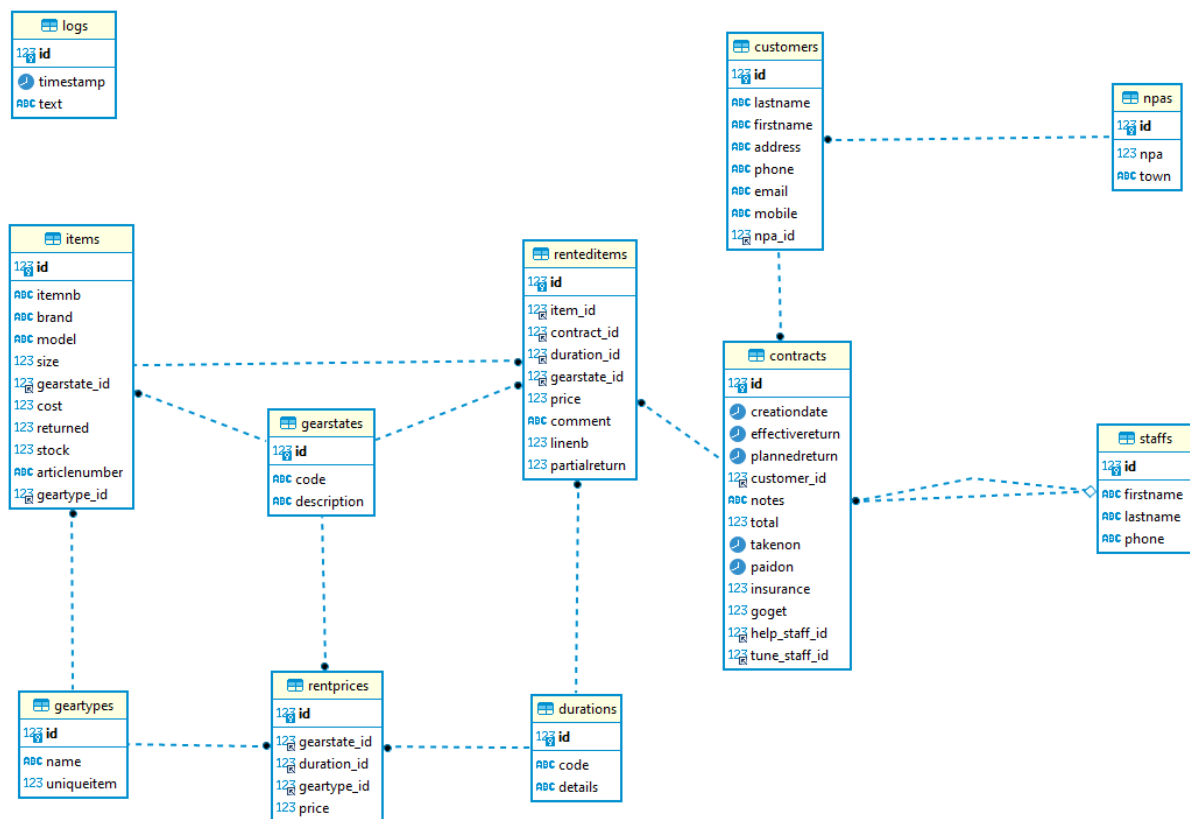
MCD SPLOKS, 13 mai 2021



### 2.1.4 Modèle logique des données

Le modèle logique des données, MLD, a pour but décrire la structure des données utilisées sans faire référence à un langage de programmation. Il précise quel type de données sont utilisées lors du traitement (Texte, Int, Float, date, etc...).

Ceci a été créé à la suite d'avoir validé le MCD avec le client.



MLD SPLOKS, mai 2022

## 2.2 Stratégie de test

Pour le développement de SPLOKS, la base de données MySQL sera installée sur le même PC que celui sur lequel j'écrirai le code. Sur ce poste, je ne ferai que des tests unitaires du type CRUD (Create, Read, Update, Delete) sur les différentes Classes existantes.

Je préparerai :

- Un script par Classe qui va créer, lire, modifier et effacer un nouvel objet de chaque classe.

Ensuite, j'utiliserai 5 PC simultanément pour tester la robustesse de SPLOKS. Le but est de simuler l'environnement du magasin Sports-Time d'Echallens (regarder le schéma à la page 7) et vérifier le bon fonctionnement de SPLOKS quand plusieurs machines manipulent les données au même temps. Les machines seront toutes du Windows 10 (comme pour le magasin Sports-Time d'Echallens), elles seront toutes connectées à un partage de connexion 4G.

Pour tout ce qui est données, j'utilise une base MySQL contenant environ 20'000 entrées différentes. Cela est une quantité assez importante de données et permettra de vérifier la performance de SPLOKS face à cela.

## 2.3 Risques techniques

### 2.3.1 GitHub

On utilise GitHub pour stocker notre projet en acceptant le risque qu'un jour, ou un moment donné, GitHub peut avoir des problèmes ou des mises à jour et qu'on se trouve du coup sans accès à notre projet pendant un moment.

GitHub apporte une grande valeur à notre projet car il nous permet de faire de faire du « Versioning » de notre projet, partager facilement les documents du projet avec les membres de l'équipe/client et aussi de stocker notre projet dans le cloud afin d'avoir accès depuis n'importe quelle machine avec une connexion internet.

Il nous aide aussi avec le prochain point.

### 2.3.2 Problème avec disque de stockage

Si le disque HDD ou SSD de ma machine ne fonctionne plus, je peux facilement contourner le problème à l'aide de GitHub.

GitHub me permet très rapidement de télécharger la dernière version mise sur le serveur. Cela permet de limiter les pertes (à la dernière version postée).

Avec cela, on accepte que le projet est stocké sur un serveur externe (GitHub) et qu'on est dépendant de leurs choix/problèmes (dates de mises à jour, problèmes de serveur, etc...).

---

## 3 Réalisation

### 3.1 Erreurs restantes

#### 3.1.1 Trop d'items à la génération d'un contrat PDF

Lors de la création d'un contrat, quand on appuie sur le bouton imprimer avec plus de 8 items, le contenu final du contrat est écrasé par les notes de bas de page du contrat.

### 3.2 Liste des documents fournis

#### 3.2.1 Journal de travail

Le journal de travail peut être trouvé directement sur le contenu du projet. Cela se trouve dans le dossier « logbook » à la racine du projet. À l'intérieur, il se trouve le journal de bord, appelé « sploks\_logbook.md », ainsi qu'un extrait du journal de travail par sprint, nommé de la façon suivante :

sprint\_xx\_worktime.pdf

Les « xx » étant le numéro du sprint, par exemple : « sprint\_01\_worktime.pdf ».

#### 3.2.2 Journal de bord

Le journal de bord peut être trouvé directement sur le contenu du projet. Cela se trouve dans le dossier « logbook » à la racine du projet, sous le nom de : « sploks\_logbook.md ». Le journal de bord contient uniquement les événements significatifs au long du développement du projet.

#### 3.2.3 Documentation de projet

C'est ce fichier que vous êtes en train de lire actuellement. Il contient les informations sur tout le projet, pourquoi ce projet, pourquoi Python, etc...

Il se trouve dans le dossier « docs » à la racine du projet. Deux versions sont disponibles : une version PDF et une version modifiable avec Word.

#### 3.2.4 Résumé du rapport

C'est un résumé conceptuel du travail effectué et du résultat obtenu. Il permet très rapidement de comprendre le travail fourni.

## 4 Conclusions

### 4.1 Objectifs atteints

Objectif	Commentaire
Affichage de la liste de contrats	Une fenêtre contenant la liste de tous les contrats a été créée et est totalement fonctionnelle.
Afficher uniquement les contrats en retard sur la liste de contrats	Quand on coche la checkbox « En retard » dans la liste de contrats, sont affichés uniquement les contrats en retard.
Uniformisation du code	Retravaillé l'affichage de la liste clients, instauré le champ de recherche permettant de filtrer la liste de clients. Retravaillé l'affichage de la liste de contrats et instauré le champ de recherche permettant de filtrer la liste de clients
Affichage de contrats d'un article/client	Quand on ouvre l'inspecteur d'un article/client, un bouton « liste de contrats » est actif si l'article/client possède des contrats à son nom. En appuyant dessus, une fenêtre s'ouvre avec la liste de contrats.
Ajuster le prix d'un item	Au moment d'ajouter un item sur le contrat de location, le prix s'ajuste à l'état de l'item et à la durée de location. Si un prix n'a pas été défini, le prix zéro est affiché. On peut aussi introduire un prix à notre choix.
Générer un contrat	Il est possible de créer un contrat. Le contrat se construit au fur et à mesure qu'on remplit les champs. Au début, on choisit un client, ensuite on a la possibilité d'ajouter des articles à la liste d'articles à louer. On bloque la liste, il ne faut plus qu'ajouter les dates et cliquer sur le bouton conclure.
Imprimer un contrat	Après avoir conclu un contrat, il suffit d'appuyer sur le bouton imprimé et un contrat PDF est généré automatiquement.

## 4.2 Objectifs non-atteints

Objectif	Commentaire
Gérer l'état des contrats de location	Il n'est pas possible de clôturer un contrat du contrat après l'avoir créé.
Effectuer la stratégie de tests	Je n'ai malheureusement pas pu effectuer des tests, mon temps a été totalement consacré à développer les fonctionnalités demandées et ne m'a pas permis de tout faire.

## 4.3 Points positifs / négatifs

### 4.3.1 Positifs

Je suis content du résultat actuel de l'application, j'ai pu effectuer des choses que je ne pensais jamais faire :

- Un filtre à 8 champs (Filtre pour sélectionner le client dans la création d'un contrat)
- Générer un PDF avec Python
- Etc...

SPLOKS arrive gentiment sur un point final, il reste très peu de choses à corriger/terminer. Je trouve que l'application respecte ce qui a été demandé tout au début par Monsieur Carrel, c'est-à-dire, de rester assez proche de Coliks et surtout de façon très intuitive et performante.

### 4.3.2 Négatifs

Je suis déçu de ne pas avoir arrivé à respecter entièrement mon cahier des charges, même s'il est chargé. SPLOKS est un projet qui me tiens à cœur, c'est mon premier vrai projet et surtout ma première application en Python.

J'aurai aimé avoir un peu plus de temps, pour effectuer ma stratégie de tests et arrivé avec un produit qui a été testé et dont les erreurs ont été documentées.

## 4.4 Suites possibles pour le projet

Le projet doit continuer, comme cité plus haut, le projet est en phase finale. Il ne reste plus énormément de chose à ajouter, il reste surtout des choses à corriger. Je pense qu'une première présentation au manager de Sports-Time d'Echallens devrait arriver bientôt afin de confirmer son intérêt par SPLOKS et avoir son premier feedback.

## 5 Annexes

### 5.1 Sources – Bibliographie

#### 5.1.1 Doutes MySQL

Pendant le projet, j'ai été mené à écrire des commandes SQL pour amener les données de la base de données à l'application Python. À l'aide de la section SQL du site W3Schools j'ai pu avoir un support à consulter pour m'aider à écrire mes fonctions.

W3Schools contient un bon support basique pour plusieurs langages, notamment le Python que j'ai pu aussi consulter.

Lien : <https://www.w3schools.com/>

#### 5.1.2 Tutoriels Python

Pendant le développement du projet, je suis tombé à plusieurs reprises sur ce site appelé « Real Python ». Ils proposent énormément de contenu et d'informations sur le Python. On peut y trouver aussi énormément de tutoriels et de conseils.

Lien : <https://realpython.com/>

#### 5.1.3 Utilisation de Reportlab

Pour générer un fichier PDF à l'aide de Python, j'ai trouvé ce petit tutoriel de débutant sur la librairie Reportlab. Je me suis beaucoup basé sur les exemples donnés pour développer notre fonctionnalité de génération de contrat.

Lien : <https://www.blog.pythonlibrary.org/2010/03/08/a-simple-step-by-step-reportlab-tutorial/>

### 5.2 Journal de travail

Le journal de travail peut être trouvé directement sur le contenu du projet. Cela se trouve dans le dossier « logbook » à la racine du projet. À l'intérieur, il se trouve le journal de bord, appelé « sploks\_logbook.md », ainsi qu'un extrait du journal de travail par sprint, nommé de la façon suivante :

*sprint\_xx\_worktime.pdf*

Les « xx » étant le numéro du sprint, par exemple : « sprint\_01\_worktime.pdf ».

### 5.3 Manuel d'Installation

Un fichier « README.md » se trouve à la racine du projet. À l'intérieur, vous allez trouver une liste des applications que nous avons utilisé, ainsi qu'une marche à suivre de l'installation pour pouvoir lancer le projet.

## 5.4 Manuel d'utilisation

Il contient un guide d'utilisation de l'application. Il explique les particularités de chaque interface, ses fonctionnalités et les raccourcis que vous pouvez utiliser. Il se trouve dans le dossier « docs » à la racine du projet. Deux versions sont disponibles : une version PDF et une version modifiable avec Word. Les deux sont nommé « USERGUIDE ».

Attention, ce manuel n'est pas à jour avec la version actuelle de SPLOKS. Ce manuel date de l'avant TPI donc il y a certaines fonctionnalités qui ne sont pas citées dans le document.