



SPLOKS

Projet TPI 4^{ème} année

Résumé

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

<p>Auteur : Marwan Alhelo E-mail : marwan.alhlo@cpnv.ch Classe : SI-C4a Date :13.05.2022</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.....	9
2.3	Risques techniques	9
2.4	Planification	9
2.5	Dossier de conception	9
3	Réalisation.....	10
3.1	Dossier de réalisation	10
3.2	Description des tests effectués	10
3.3	Erreurs restantes	10
3.4	Liste des documents fournis	11
4	Conclusions	11
5	Annexes.....	12
5.1	Résumé du rapport du TPI / version succincte de la documentation	12
5.2	Sources – Bibliographie	12
5.3	Journal de travail	12
5.4	Manuel d'Installation	12
5.5	Manuel d'Utilisation.....	12
5.6	Archives du projet.....	12

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.

1.2 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,

1.2.1 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'affaire (en matière de location) réalisé avec ce client
- S'il a du matériel à retourner en retard

1.2.2 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).

1.2.3 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.

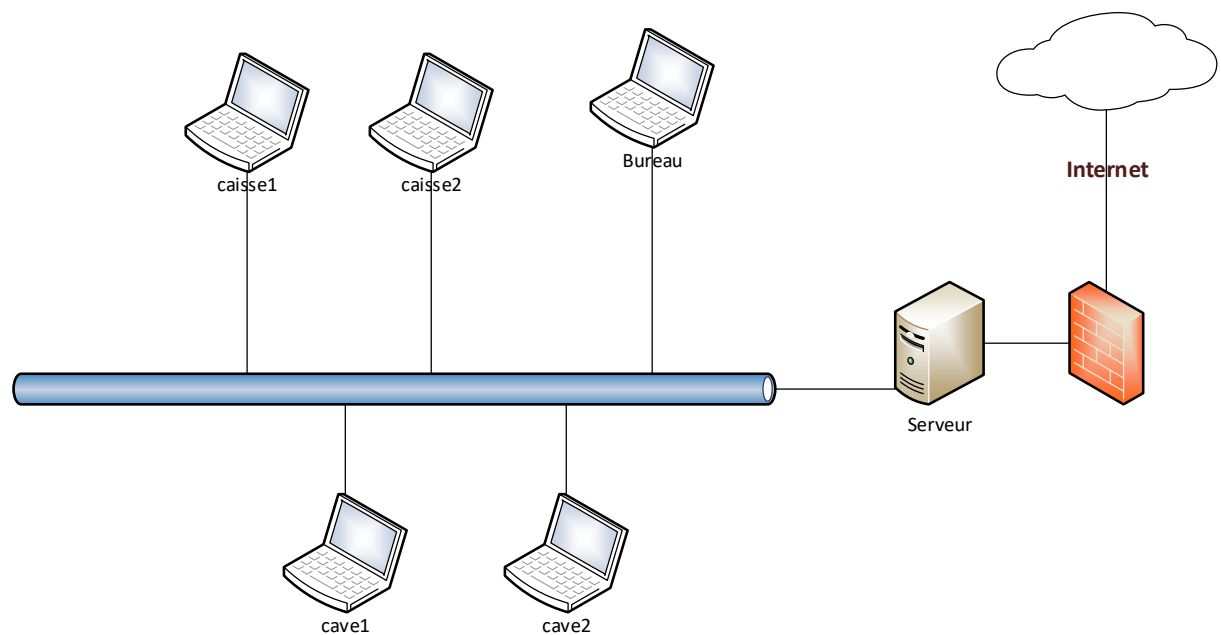
1.3 Planification initiale

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		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%			

2 Analyse / Conception

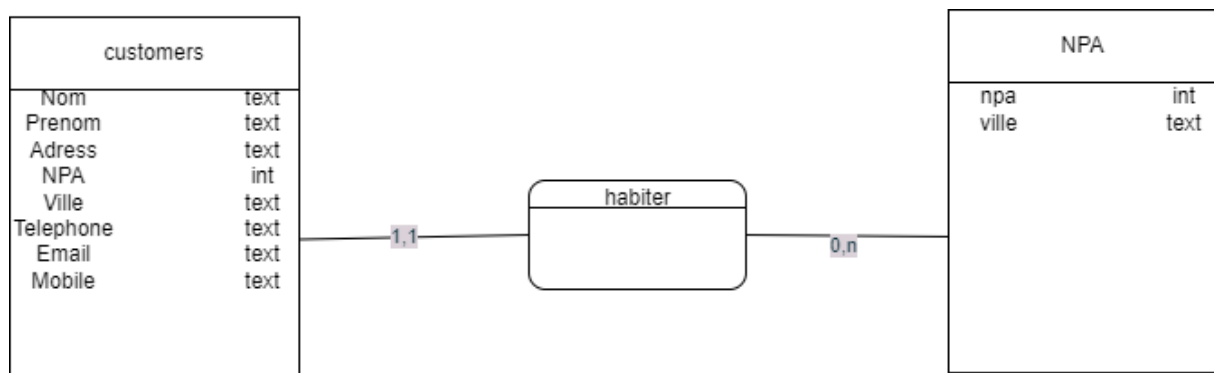
2.1 Concept

- 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



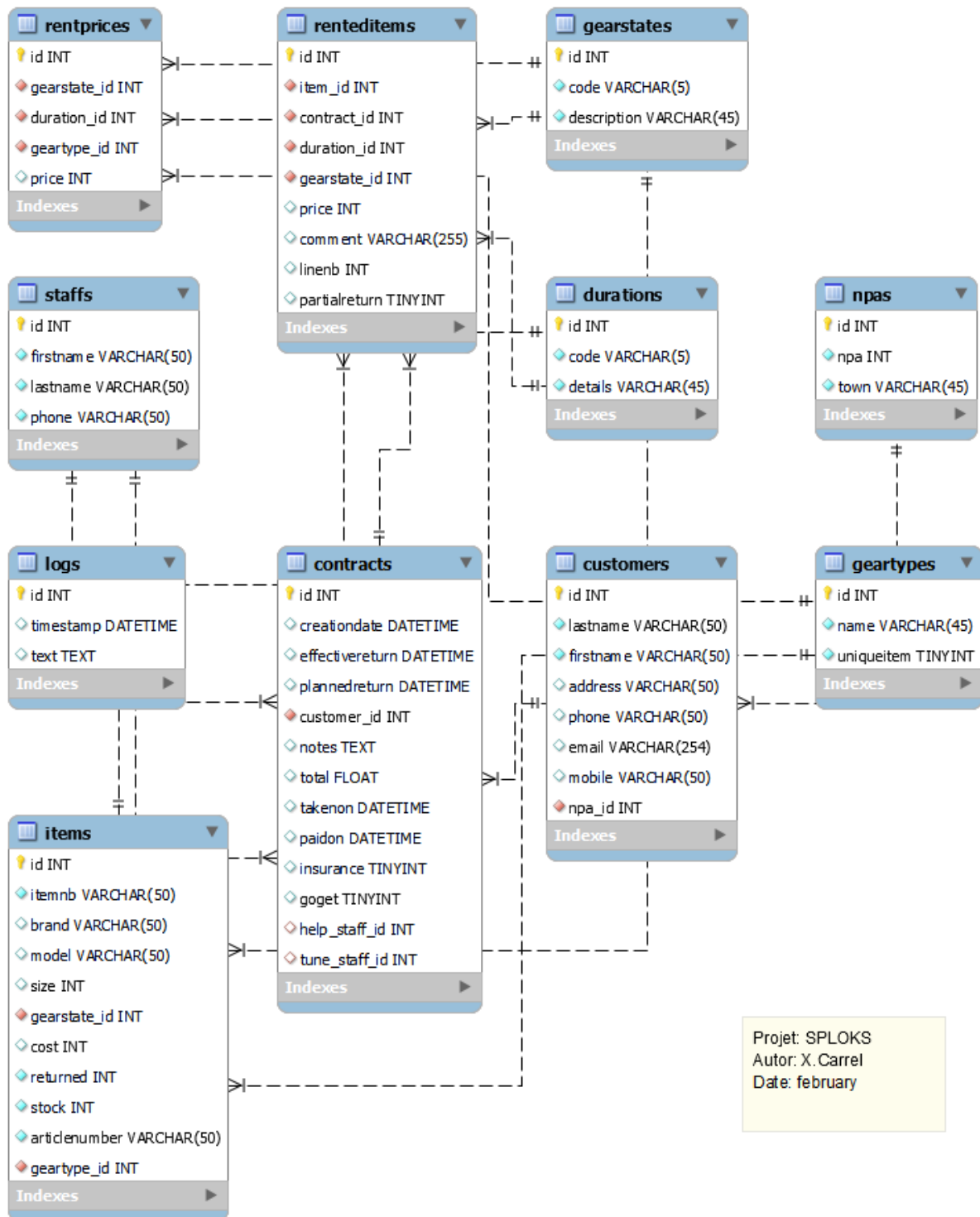
2.2 Modèle conceptuel de données (MCD)

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.



2.3 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



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...

- ...

2.4 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.

2.5 Risques techniques

- risques techniques (complexité, manque de compétences, ...).

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

2.6 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.*

2.7 Dossier de conception

Fournir tous les document de conception:

- le choix du matériel HW
- le choix des systèmes d'exploitation pour la réalisation et l'utilisation
- 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...*

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

3 Réalisation

3.1 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 : 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...

3.2 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*

3.3 Erreurs restantes

S'il reste encore des erreurs:

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

3.4 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...*

4 Conclusions

Développez en tous cas les points suivants:

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

5 Annexes

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

5.2 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)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4

5.5 Manuel d'Installation

5.6 Manuel d'Utilisation

5.7 Archives du projet

Media, ... dans une fourre en plastique