

Projet Rack à Clés

Created by Jonathan Debétaz

- [1. Analyse préliminaire](#)
 - [1.1 Introduction](#)
 - [1.2 Organisation](#)
 - [1.3 Objectifs](#)
 - [Use case 1: Le secrétariat gère le stock de clés](#)
 - [Scénario: Consultation du stock](#)
 - [Scénario: Elimination du stock](#)
 - [Scénario: Ajout au stock](#)
 - [Use case 2: Un enseignant utilise des clés](#)
 - [Scénario: Réservation](#)
 - [Scénario: Libération](#)
 - [1.4 Planification initiale](#)
- [2. Analyse](#)
 - [2.1 Cahier des charges détaillé](#)
 - [2.1.1 Définition du contenu et des fonctionnalités](#)
 - [2.1.2 Analyse concurrentielle](#)
 - [2.1.3 Architecture de l'application](#)
 - [2.1.4 Maquette préliminaire](#)
 - [2.1.5 Nom du site et nom de domaine](#)
 - [2.1.6 Référencement](#)
 - [2.1.7 Définition de l'audience](#)
 - [2.1.8 Définition des intervenant](#)
 - [2.2 Stratégie de tests](#)
 - [2.3 Budget initial](#)
 - [2.4 Planification](#)
 - [2.5 Historique](#)
- [3. Conception](#)
 - [3.1 Analyse de l'environnement](#)
 - [3.2 Détermination de l'arborescence du site et des rubriques](#)
 - [3.3 Définition de la charte graphique](#)
 - [3.4 Maquette graphique](#)
 - [3.5 Conception de la base de donnée](#)
 - [3.6 Conception du code](#)
 - [3.7 Plugins](#)
 - [3.8 Choix de la formule d'hébergement](#)
- [4. Réalisation](#)
 - [4.1 Dossier de réalisation](#)
 - [4.2 Description des tests effectués](#)
 - [4.3 Erreurs restantes](#)
 - [4.4 Dossier d'archivage](#)
- [5. Mise en service](#)
 - [5.1 Rapport de la mise en service](#)

- [5.2 Liste des documents fournis](#)
- [6. Conclusions](#)
- [7. Annexes](#)
 - [7.1 Sources - Bibliographie](#)
 - [7.2 Journal de bord de chaque participant](#)
 - [7.3 Manuel d'installation](#)
 - [7.4 Manuel d'utilisation](#)
 - [7.5 Archive du projet](#)

1. Analyse préliminaire

1.1 Introduction

Durant les semaine d'examens, les enseignants de la filière informtique du CPNV utilisent fréquemment des clés USB pour distribuer des données nécessaires à l'examen et/ou pour collecter le travail effectué par les élèves. Cela donne lieu à passablement de manipulations dont on aimerait se passer.

Le but de ce projet est d'offrir une interface permettant de gérer un parc d'une quinzaine de clés USB à travers plusieurs hubs USB multiports

1.2 Organisation

- Elève: Jonathan Debétaz, jonathan.debetaz@cpnv.ch
- Chef de projet: Cédric Rotten, cedric.rotten@cpnv.ch
- Expert: Xavier Carrel, xavier.carrel@cpnv.ch

1.3 Objectifs

Use case 1: Le secrétariat gère le stock de clés

Scénario: Consultation du stock

Le secrétariat – ou toute personne ayant accès – ouvre l'application et consulte le tableau, qui montre la liste

des clés du stock, avec pour chacune:

- Son état
 - Disponible
 - Présente
 - Absente
 - Utilisée
- Le port du rack où elle se trouve (si l'état est Disponible ou Présente)
- La personne qui l'a réservée (si l'état est Présente ou Utilisée)
- Son espace disponible

Scénario: Elimination du stock

Une clé est absente et introuvable, ou on n'arrive plus à la réinitialiser => le secrétariat l'élimine du stock

Scénario: Ajout au stock

Pour compenser les pertes, le secrétariat veut rajouter des clés dans le stock. Il met des clés vierges et inconnues du système dans des ports libres et lance la procédure d'initialisation. Le système détecte les nouvelles clés et les initialise.

Use case 2: Un enseignant utilise des clés

Scénario: Réservation

Le prof définit une nouvelle réservation:

- Un nom (par exemple "ICT 431")
- Le nombre de clés voulues
- Le contenu de la clé. Uniquement des archives compressées

Lorsqu'il sauve la réservation, le système vérifie que le nombre de clés disponibles ainsi que l'espace restant est suffisant pour satisfaire la demande.

Si tel est le cas, le système charge les clés et les marque comme Présente.

Scénario: Libération

Une fois l'examen terminé, le prof remet les clés dans le rack, retourne sur sa réservation, indique un dossier dans lequel placer les données récupérées et lance la récupération. Le système recherche les clés de la réservation dans le rack. Pour chaque clé trouvée, il transfère le contenu de la clé dans le dossier de récupération et la réinitialise. Il libère ensuite la clé. Si toutes les clés de la réservation ont été libérées, la réservation est marquée comme terminée.

1.4 Planification initiale

2. Analyse

2.1 Cahier des charges détaillé

2.1.1 Définition du contenu et des fonctionnalités

2.1.2 Analyse concurrentielle

????

2.1.3 Architecture de l'application

- client

- **controller** Controller for AngularJS
- **directive** Directive for AngularJS
- **service**
- **static** Public directory access
 - **css**
 - **fonts**
 - **img**
 - **js**
- **views** Views directory for AngularJS
- **config** Configuration directory for Application
- **model** Models directory for MongoDB storage system
- **scripts**
 - **batch** batch system
 - **usb** scripts for usb events
- **server**

2.1.4 Maquette préliminaire

Liste des clés

Faire une réservation de clés

2.1.5 Nom du site et nom de domaine

Le site portera le nom `Rack à clés`, et le nom de domaine et encore inconnu et indéci.

2.1.6 Référencement

Le site ne sera référencé sur aucun moteur de recherche, il sera uniquement ajouter au portail de l'intranet.

2.1.7 Définition de l'audiance

Seul les enseignants et l'équipe pédagogique pourront avoir accès à l'application.

2.1.8 Définition des intervenant

2.2 Stratégie de tests

2.3 Budget initial

Quantité	Matériel	Prix
2	Hub USB ExSys Ex-1178	41.45
1	RaspberryPi 3b	50.20
Total		132.10

2.4 Planification

2.5 Historique

L'historique est disponible en tout temps sur [le compte github du CPNV](#)

3. Conception

3.1 Analyse de l'environnement

3.1.1 Environnement de développement

- Server MongoDB Base de donnée NoSQL
- NodeJS Interpréteur JavaScript côté serveur
- ArchLinux Système d'exploitation
 - commande `blkid` listage des peripheriques de stockage disponible sur le système d'exploitation (execution en root)
- Windows Serveur 2012 Serveur de gestion des compte AD

3.1.2 Environnement de production

- Serveur Nginx Serveur Web
- Server MongoDB Base de donnée NoSQL
- NodeJS Interpréteur JavaScript côté serveur
- Debian Système d'exploitation
 - commande `blkid` listage des peripheriques de stockage disponible sur le système d'exploitation (execution en root)
- Windows Serveur Serveur de gestion des compte AD

3.2 Détermination de l'arborescence du site et des rubriques

- Liste des clés USB
 - Réserver des clés USB
- Mes réservations
- Déconnexion
- Connexion
- Administration
 - Liste des réservations

3.3 Définition de la charte graphique

Reprise de la charte graphique du CPNV

3.4 Maquette graphique

3.5 Conception de la base de donnée

MLD-MCD

3.6 Conception du code

3.7 Plugins

- [AngularJS 1](#)
- [Semantic UI](#)

3.8 Choix de la formule d'hébergement

L'hébergement du site sera sur un Raspberry PI

4. Réalisation

4.1 Dossier de réalisation

4.2 Description des tests effectués

4.3 Erreurs restantes

4.4 Dossier d'archivage

Une archive est disponible en tout temps sur [le compte github du CPNV](#)

5. Mise en service

5.1 Rapport de la mise en service

5.2 Liste des documents fournis

6. Conclusions

7. Annexes

7.1 Sources - Bibliographie

7.2 Journal de bord de chaque participant

[Lien vers le journal de bord de Jonathan Debétaz](#)

7.3 Manuel d'installation

7.4 Manuel d'utilisation

7.5 Archive du projet

Une archive est disponible en tout temps sur [le compte github du CPNV](#)