

Active

– *Still active* –



@ Auteurs

NDJANDA MBIADA Jacques Charles¹
LEBEAU Christophe
JOOSSEN Matthieu

@ Mentors

ZACCHIROLI Stefano
KESNER Delia
SIGHIREANU Mihaela

¹ Chef de projet.

SOMMAIRE

- [Préface](#)
 - [Document](#)
 - [Versions](#)
- [Introduction](#)
 - [Les besoins](#)
 - [Brève description](#)
 - [Objectifs du produit](#)
- [Charges d'utilisation](#)
 - [Généralité](#)
 - [Public visé](#)
 - [Pré-requis](#)
 - [Spécification système des utilisateurs](#)
 - [Cas d'utilisation](#)
 - [Interaction des utilisateurs](#)
- [Architecture du système](#)
 - [Vue d'ensemble](#)
 - [Modularité](#)
- [Charges du système](#)
- [Modèles du système](#)
- [Evolution du systeme](#)
 - [Évolution des besoins](#)
 - [Évolution matériel](#)
 - [Mise à jour](#)
 - [Mise à jour planifiée](#)
 - [Mise à jour conflictuel](#)
- [Annexes](#)
 - [Hardware requirement](#)
 - [Linux / GNU Unix](#)
 - [Sqlite 3](#)
- [Index](#)
 - [Active Query](#)
 - [Termes](#)
 - [Modules](#)
 - [Diagrammes](#)
 - [Fonctions](#)

— Préface

Document

Ce document a pour but de décrire les besoins, les exigences, les contraintes et les limites que nous allons devoir respecter durant la réalisation de notre projet.

Il résume les tâches que le programme doit pouvoir effectuer, Le document est composé de d'une première partie ouverte à tout lecteur et d'une partie technique qui explique les spécificités du produit plutôt destinée à un utilisateur averti.

Le document sert également de planning pour délimiter le temps nécessaire à la réalisation de chaque étape du projet et ce dans le but de le terminer à temps.

Versions

Nous souhaitons présenter ci-dessous l'intérêt de chaque version de ce présent document. Les modifications et ajouts importantes.

Version 2

Finalisation de la spécification en vu du début de développement.

Définition du langage de requête et liste des charges.

Version 2 Beta ← (HEAD)

Définition du langage de requête et support de définition. Choix final de la structure global

Des langages et technologies utilisés.

Version 2 Alpha

Restructuration du document. Avec respect du modèle donné en cours

Version 1

Brouillon complet avec éléments de spécification et de la structure global du projet

Version 1 Alpha & Beta

Brouillon du cahier de charge, esquisse des

— Introduction

[Indiquer à quel besoin le produit répond. Décrire brièvement les fonctionnalités du produit et ses interactions avec d'autres systèmes. Positionner le produit par rapport à la stratégie ou les objectifs du client.]

Les besoins

Le progrès technologique est indéniable à notre époque, notamment en termes de capacité de stockage des ordinateurs personnels. L'un des enjeux des systèmes pour PC est la gestion des données utilisateur, à savoir permettre à ces derniers d'accéder rapidement à une information datant de plusieurs semaines, mois ou années sans y perdre des heures de recherche.

La capacité de nos disques durs peut atteindre aujourd'hui des centaines de gigaoctets (Go) et nous pouvons donc stocker des millions de fichiers sur un ordinateur personnel.

Un gestionnaire de système de fichier est un programme permettant d'accéder à ces données, d'y naviguer et de visualiser leur relation (quel fichier appartient à quel dossier par exemple). Ce programme n'a pas pour rôle de trier et de rechercher les fichiers et c'est pourquoi en l'état, l'utilisateur devra trier son disque dur à la main. Or, non seulement la machine sait mieux trier et rechercher que nous mais en plus elle le fait bien plus rapidement en cas d'oubli.

Bien que la plupart des systèmes d'exploitation répandus disposent en plus du gestionnaire de système de fichier un programme d'indexation, nous souhaitons écrire un programme alternatif au programme d'indexation natif de Linux / Unix et pourquoi ne pas le surpasser sur le plan technique et esthétique.

La réalisation du projet ne doit nécessiter aucun fond d'investissement, user uniquement de technologies open sources ou libres d'utilisation sans contraintes particulières sur le produit final. Pour palier à tout dépense supplémentaire (notamment sur le matériel de travail), nous avons à disposition des machines sous différentes plates-formes (NT, Linux, Unix) connectées à internet en très haut débit. Tout module permettant la réalisation du projet est inexistant en utilisation libre et devra alors être développé par l'équipe en charge du projet. Le but est bien entendu de rendre le programme terminé, mais également de parvenir à produire ce dernier par une démarche professionnelle et de manière coopérative sous la direction d'un chef de projet. — La présentation du projet est faite ci-dessous. —

Brève description

Objectifs du produit

— Charges d'utilisation

[Décrire les fonctionnalités offertes à l'utilisateur ainsi que des charges non fonctionnelles du système. Cette description peut utiliser le langage naturel, des diagrammes ou d'autres notations compréhensibles par les clients. Les standards qui doivent être respectés par le produit doivent être spécifiés.]

Généralité

Public visé

[Souligner le public visé en décrivant pourquoi et comment cela leur est utile]

Pré-requis

[Décrire les pré requis tant au niveau système (petit résumé de la partie Annexe:Hardware requirement & linux/gnu unix) que compétences de l'utilisateur]

Spécification système des utilisateurs

Cas d'utilisation

[Préciser comment est prévu l'utilisation du programme par l'utilisateur]

[diagrammes d'utilisation]

préciser comment est l'interface générale, ce que l'on peut faire comme opération.

Interaction des utilisateurs

[expliciter les interactions permises à l'utilisateur sur le programme et entre utilisateurs (dans notre cas pas possible apparemment)]

[diagramme d'interaction]

Préciser sur le résultat attendu de l'interface par les actions effectuées.

Les vues produits etc.

Indiquer que plusieurs utilisateurs peuvent utiliser le système simultanément.

— Architecture du système

[Présenter une vue de haut niveau de l'architecture preconisee du systeme et la distribution de fonctionnalites a travers les modules du systeme. Les composantes

reutilisee de l'architecture doivent être soulignees.

Cette partie est incontestablement la plus longue il faut donc bien la réfléchir notamment ce qu'elle ne devrait pas contenir !]

Vue d'ensemble

Modularité

— Charges du système

[Décrire les charges fonctionnelles en détail et d'autre charges non fonctionnelles, par exemple donner des détails sur les charges d'interface avec d'autres systemes.]

Généralité

— Modèles du système

[Donner un ou plusieurs modeles du systeme et montrer la relation entre les composantes du systeme et son environnement.]

— Evolution du système

[Decrire les hypotheses fondamentales sur lesquelles le systeme a ete construit et anticiper les changements dus a l'evolution du materiel, aux changements des besoins d'utilisations, etc.]

Évolution des besoins

Évolution matériel

Mise à jour

Mise à jour planifiée

Mise à jour conflictuel

— Annexes

[Fournir des informations détaillées et spécifiques au produit développé. Exemples d'annexes qui peuvent être données : description du matériel et de la base de données utilisée. Pour le matériel, définir la configuration minimale et optimale pour utiliser le produit. Pour la base de données, définir le modèle relationnel.]

Hardware requirement

Linux / GNU Unix

Sqlite 3

— Index

[Plusieurs indexations du document peuvent y paraître : l'index des termes utilisés, l'index des diagrammes, l'index des fonctions, etc.]

Active Query

Active Query est le langage permettant de communiquer avec le programme. Ce langage de requête permet de mettre à jour la base de données et d'y effectuer des recherches.

Définition (AQ) :

AQ ::= [[1-9]Digit*] R [C R]*

R ::= #A Q [C Q]*

A ::= I | S | U | D | “”

Q ::= [O.]{F}[.T]

O ::= core | irl

T ::= text|video|sound|object|archive|other

F ::= **Field**:“**Value**”[C F]

C ::= “ ” | ,

Field ::= s | r | type |author|dlm|dlo|dc|title|album|singer|style|genre|content|...

Value ::= **Regex** | JSON | XML

String r = “#I irl.{(name content):str content:str2}.audio {author:str}.audio#S irl.{(name content):str content:str2}.audio {author:str}.audio#S irl.{(name content):str content:str2}.audio {author:str}.audio#D irl.{(name content):str content:str2}.audio {author:str}.audio#S irl.{(name content):str content:str2}.audio {author:str}.audio#S irl.{(name content):str content:str2}.audio {author:str}.audio”

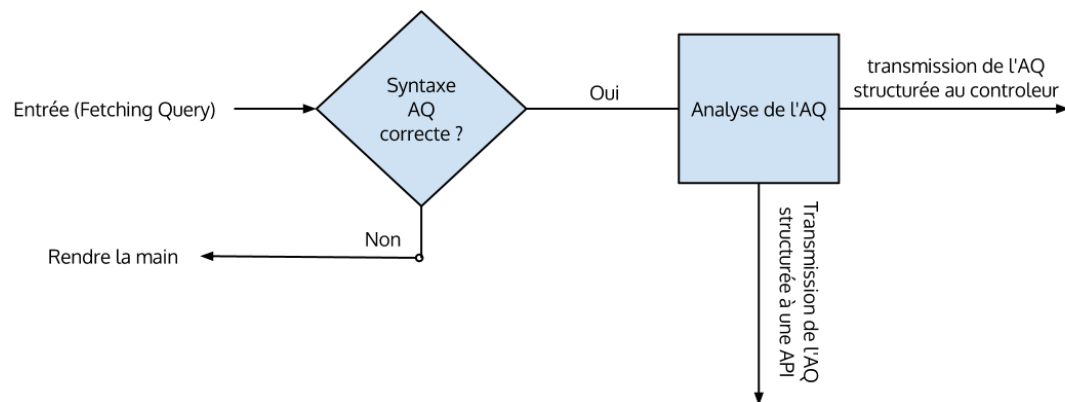
Termes

Modules

Diagrammes

Fonctions

AQ Analyser



© Active