|  |
| --- |
|  |
| Syntaxi |
| Extracteur de documentation |
|  |
| **Thomas Auguey** |
| **22/10/2015** |

|  |
| --- |
| Système de documentation. |

SOMMAIRE

[Présentation 3](#_Toc434340970)

[Motivations 3](#_Toc434340971)

[Cahier des charges 3](#_Toc434340972)

[Eléments 3](#_Toc434340973)

[Acteurs 4](#_Toc434340974)

[Cas d’utilisations 5](#_Toc434340975)

[Définition 6](#_Toc434340976)

[Système 6](#_Toc434340977)

[Réalisation 7](#_Toc434340978)

[Base de données 8](#_Toc434340979)

[Interface de gestion 11](#_Toc434340980)

# Présentation

Syntaxi est un gestionnaire de documentation capable d’extraire la documentation de fichiers textes dans des objets de données. Ces données sont ainsi exploitables par des applications tiers pour mettre en forme des documentations utilisateurs. L’intérêt du système repose sur la possibilité pour l’utilisateur final de créer ces règles d’extraction basées sur les expressions-régulières (bien connue des programmeurs en informatique). Le projet vise donc principalement les documentations techniques et les API informatiques.

## Motivations

L’avantage est de tirer parti de la flexibilité des expressions régulières. Contrairement aux logiciels sur le marché programmé pour analyser des langages spécifiques (C++,C#,Java,…) tel que Doxygen, Syntaxi prend la problématique dans le sens inverse en proposant de s’adapter au code existant en se basant sur des expression régulières pour extraire le texte intéressant.

## Cahier des charges

Le système devra proposer un modèle de données permettant de stocker des objets de contenu texte génériques et typés. Les objets seront associés à une expression régulière permettant leurs extractions depuis un fichier texte quelconque. Les objets ainsi créés seront stocké sous forme de données persistantes lisible par une application tierce.

Le système ne définissant pas de mise en forme des objets (classe de données, page de description, fonction, élément de template, …) l’utilisateur final devra typer ses expressions dans le but de les identifier lors de la mise en page.

Idéalement, l’application devra permettre de mettre à jour des projets existant et d’exporter ses données dans une base SQL distante.

L’application sera proposée sous forme de ligne de commande et d’IHM. L’interface en ligne de commande est utilisée dans un but d’automatisation du processus de génération des données.

## Eléments

### Projet

Contient les objets de données générées et les objets de syntaxe.

### Contenu

Société assimilé au groupe. Les informations connues comprennent le minimum nécessaire à l’entreprise (raison sociale)

### Syntaxe

Il s’agit d’une tache réalisé par un employé dans le cadre de son travail. Les informations importantes sont la nature de la prestation et la date d’intervention.

### Configuration

La configuration contient les informations de connexion au serveur de données. Cette configuration est unique à chaque HIM client installé.

## Acteurs

### Utilisateur

L’utilisateur final contrôle de plein droit les données du projet à l’aide d’une IHM

## Cas d’utilisations

# Définition

## Système

### Base de données

Une base de données compatible ODBC permettra de stocker et partager les données aux clients système.

### Interface de commande

Une interface de gestion en ligne de commande permettra d’automatiser la gestion des données du projet par une tache planifié ou un utilisateur final.

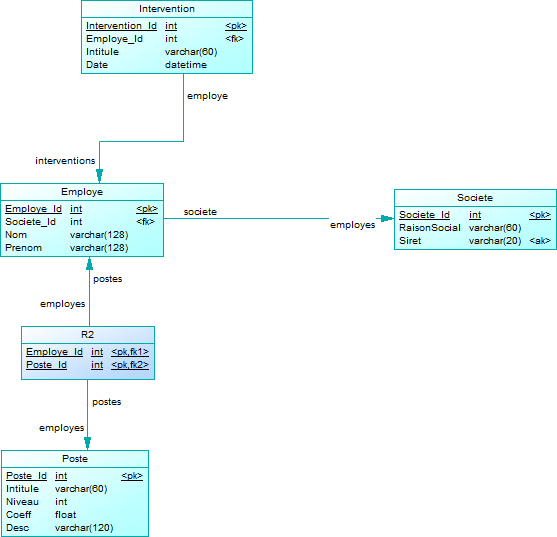
### Interface de gestion

Une interface de gestion homme-machine pour Windows permettra de gérer les données du projet par l’utilisateur final.

# Réalisation

## Base de données

### Modèle de la base de données SQL



### Tables

Définition des tables SQL

|  |  |  |
| --- | --- | --- |
| **Employe**  Employé | | |
| Code | Data Type | Comment |
| Employe\_Id | int | Identificateur |
| Societe\_Id | int | Société |
| Nom | varchar(128) | Nom |
| Prenom | varchar(250) | Prénom |
| Index | Fields | Comment |
| PK\_EMPLOYE | Employe\_Id | Clé primaire |
| FK\_EMPLOYE\_R1\_SOCIETE | Societe(Societe\_Id) | Clé étrangère de la société |
| Trigger | Condition | Comment |
| - | - | - |

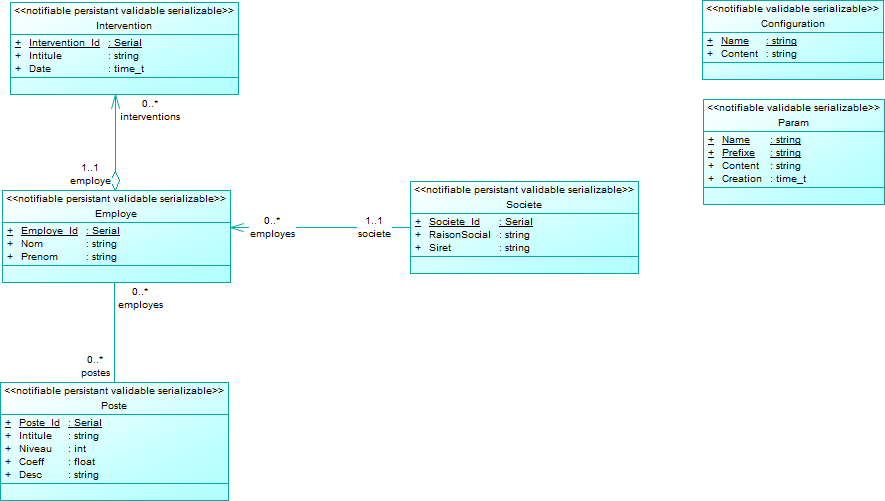
### Procédures stockées

Définition des procédures stockées SQL

## Interface de gestion

### Diagramme de classes

Model de données de l’interface utilisateur



### Entités

Définition des classes d’entités

|  |  |  |
| --- | --- | --- |
| **Employe**  Employé | | |
| Code | Data Type | Comment |
| Employe\_Id | int | Identificateur |
| Nom | string | Nom |
| Prenom | string | Prénom |
| Relation | Data Type | Comment |
| Societe | Societe | Société |