



## RAPPORT

# Mini-projet (part 2) Technos XML en binôme si4

## Rappel du contexte :

Lors du premier rendu, il était question pour nous de :

- Décrire nos choix de modélisation en langage naturel.
- Écrire un schéma XML qui représente notre modélisation.
- Écrire un document XML valide par rapport à ce modèle, contenant un extrait représentatif de la base de données BigVir2084.

A présent, il est question pour nous de :

- Corriger notre ancien code notamment grâce aux remarques et commentaires faits. (Ceci sera abordé plus bas)
- Écrire (au moins) 6 feuilles de style XSLT implémentant des fonctionnalités pour l'application BigVir2084 et les Décrire en langage naturel
- Proposer une version JSON et JSON schema de votre base de données XML ; et
- Développer un code Python ou Java (ou autre) pour parser et valider vos données XML et leur appliquer les feuilles de style que vous aurez développées : dans notre cas il s'agit d'un code Python.

## Commentaires et améliorations par rapport au premier rendu :

Bon travail mais des choses sont à revoir :

**\*Il faut détailler la description d'hospitalisation et consultations :** Désormais chaque hospitalisation est caractérisée par un médecin traitant, un hôpital, une date d'entrée et de sortie ou date de consultation.

**\*Pour les tests, les attributs résultat et intitulé sont ambiguë ; et ce serait bien d'avoir des valeurs booléennes ou numériques pour pouvoir faire des calculs dessus.**

Dans tests, ces attributs ont été remplacés par la nature du test, le type du test

et son résultat est exprimé par une valeur booléenne.

**\*Il n'y a jamais plus d'une consultation ou hospitalisation dans la base ce qui n'est pas réaliste.**

Nous avons garni davantage notre base de données.

**\*Revoir encoding en utf8 et sans accent sur les noms de balises. Ceci a été corrigé**

**\*Pas de définition de types simples**

Dans cette deuxième version, nous utilisons des types simples notamment pour limiter le type de lieu (musée, plage, salle de spectacle... ou même encore le type de maladie (maladie génétique, maladie cardiovasculaire, ....). Les types simples nous ont permis de définir des contraintes de valeurs sur certains éléments.

### CAS xslt : fonctionnalités gérées :

Fonctionnalités décrites par nos 6 feuilles de style sont les suivantes :

- Transformation 1 : Notre première transformation classe les citoyens selon le nombre d'hospitalisations par ordre croissant, si le nombre d'hospitalisation est supérieur à 1, on affiche les hospitalisations de la plus récente au plus ancien
- Transformation 2 : Il est possible pour certaines maladies, de retrouver le pourcentage de personne touchées. Cette transformation, calcul donc (dans le cas du cancer du sein et de l'Hémochromatose) le pourcentage de citoyens atteints.
- Transformation 3 : Cette transformation permet de recenser dans distinctement, le nombre de citoyens ayant rencontré un médecin dans le cadre d'une consultation, ou dans le cadre d'une hospitalisation.

- Transformation4 : Cette transformation calcule pour chaque différent lieu le nombre de citoyens qui l'ont visité. Ceci pourrait se révéler utile notamment pour l'évaluation de l'affluence dans un certain lieu.
- Transformation5 : Cette transformation recense uniquement les lieux qui ont été visités et fournies une liste des citoyens(visiteurs) pour chaque lieu.
- Transformation6 : Cette transformation calcule le pourcentage des citoyens vaccinés et non vaccinés.