CONCEPTION ET REALISATION D’UNE APPLICATION WEB D’ASSOCIATION FORMA GREEN

**2.4 Gestion des risques**

La mise en place d'un tel système est toujours risquée, Forma-Green se rassure :

• Produire une documentation avec la description de l'architecture mise en place...

• Faites une liste de tous les risques qui peuvent survenir, comment vous avez évité certains d'entre eux, comment vous en avez surmonté d'autres, qui serait en charge de ces risques.

• Vous pouvez effectuer un test d'intrusion de votre système.

**Spécification et documentation (50 points)**

• Compléter la spécification des exigences avec votre plan de travail (10 points)

• Analyse des risques (20 points)

• Documentation utilisateur (10 points)

• Documentation technique (10 points)

**Architecture (30)**

• Architecture système sécurisée (écrite, conçue) (30 points)

**5 livrables**

Les étudiants doivent inclure les éléments suivants dans leur livraison finale :

• Une archive zip avec le code source du projet. Le code source doit également être fourni avec le système de construction utilisé (fichier de projet, outils automatiques, bibliothèques...), le cas échéant.

• Documentation du projet

o Documentation technique expliquant vos choix et/ou choix/détails de mise en œuvre comprenant notamment une carte du réseau pour l'architecture

o Manuel d'utilisation (doit être compréhensible par le client)

Le premier document est un document académique. S'adressant au lecteur en tant qu'enseignant, pas en tant que client. Ces documents peuvent être en français ou en anglais, à votre convenance.

**SOMMAIRE (Documentation technique)**

Introduction.

Technologie utilisée. x

Architecture logicielle. x

Cycle de vie de logiciel

* Modèle de cycle de vie utilisée. x
* Analyse.
* Conception.
* Intégration.
* Implémentation.
* Test unitaire.
* Mise en production.

Conclusion

**SOMMAIRE (Documentation Utilisateur) : Manuel d’utilisation**

* Sitemap (les liens du site : urls )
* Interfaces (Images, description et guide)

Pour réaliser ce projet sans problème, le choix d’un cycle de vie est nécessaire.

Mais, qu’est-ce qu’un Cycle de vie ? Le cycle de vie permet de détecter les erreurs au plus tôt et

ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés. Ici, notre

choix sur le cycle de vie se portera sur le modèle en “CASCADE”

ICI IMAGE DE CYCLE DE VIE EN CASCADE

Dans ce paragraphe, nous mettrons un point sur les outils/technologies utilisées lors de la

réalisation de notre application :

1. VISUAL STUDIO CODE

Visual Studio Code est un éditeur de code source qui peut être utilisé avec une variété de

langages de programmation, notamment Java, JavaScript, Go, Node.js et C+Visual Studio

Code prend immédiatement en charge presque tous les principaux

langages de programmation. Plusieurs d'entre eux sont inclus par

défaut, par exemple JavaScript, TypeScript, CSS et HTML, mais

d'autres extensions de langage peuvent être trouvées et téléchargées

gratuitement à partir de VS Code Marketplace.

2. MySQL

MySQL est un système de gestion de bases de données

relationnelles (SGBDR). Il est distribué sous une

double licence GPL et propriétaire. Il fait partie des

logiciels de gestion de base de données les plus utilisés

au monde.

3. Le Framework DJANGO

Django est un cadre de développement qui s'inspire du principe MVC ou MTV (la vue est

gérée par un gabarit) composé de 3 parties distinctes :

1. Un langage de gabarits flexible qui permet de générer du HTML, XML ou tout autre

format texte ;

2. Un contrôleur fourni sous la forme d'un « remapping » d'URL à base d'expressions

rationnelles ;

3. Une API d'accès aux données est

automatiquement générée par le cadre

compatible CRUD. Inutile d'écrire des

requêtes SQL associées à des formulaires,

elles sont générées automatiquement par

l'ORM.

4. JAVASCRIPT

JavaScript est un langage de programmation de scripts

principalement employé dans les pages web interactives et à ce

titre est une partie essentielle des applications web. Avec les

technologies HTML et CSS, JavaScript est parfois considéré

comme l'une des technologies coeur du World Wide Web.

5. BOOTSTRAP

Bootstrap est une collection d'outils utiles à la création du design

de sites et d'applications web. C'est un ensemble qui contient des

codes HTML et CSS, des formulaires, boutons, outils de

navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

Analyse et Conception

1. Dictionnaire de données

Qu’est-ce qu’un dictionnaire de données ?

Un dictionnaire des données est une collection de métadonnées ou de données de référence

nécessaire à la conception d'une base de données relationnelle. Il revêt une importance

stratégique particulière, car il est le vocabulaire commun de l'organisation. Il décrit des données

aussi importantes que les clients, les nomenclatures de produits et de services, les annuaires, etc.

Le dictionnaire de données que nous avions pu dégager suit à notre cahier charge pour décrire

les champs de notre base de données relationnelle et comme suivant :

ICI IMAGE DE DICTIONNAIRE DE DONNEES

2. Modèle logique de données (MLD)

Qu’est-ce qu’un MLD ?

Le modèle logique de données est une étape de la conception qui consiste à décrire la structure

des données utilisées sans faire référence à un langage de programmation. Il s'agit de préciser le

type de données utilisées lors des traitements.

Ici dans notre cas, on a pu élaborer le modèle logique de donnée comment suit:

ICI IMAGE DE MODELE LOGIQUE DE DONNEES (MLD)

Architecture et diagrammes

Architecture logicielle

Django utilise l'architecture MVT (modèle-vue-Template) qui s'inspire de MVC : Le modèle

interagit avec une base de données via un ORM. Tous les modèles sont réunis dans un fichier

python Models.py. La vue reçoit une requête HTTP et renvoie une réponse HTTP. Les vues se

trouvent dans le fichier Views.py. Le Template est un fichier HTML récupéré par la vue et

envoyé au visiteur avec les données des modèles.

ICI IMAGE Architecture MVT

Diagramme de classe

Qu’est-ce qu’un diagramme de classe ?

Le diagramme de classe représente les classes intervenant dans le système. Le diagramme de

classe est une représentation statique des éléments qui composent un système et de leurs

relations.

Chaque application qui va mettre en oeuvre le système sera une instance des différentes classes

qui le compose.

Ici, on a pu réaliser un diagramme de classe comme ceci, en respectant bien sur les contraintes et les règles de gestion qui se trouve dans le cahier de charge.

ICI IMAGE DE DIAGRAMME DE CLASSE