**SOMMAIRE**

Introduction…………………………………….……..5

Cycle de vie de logiciel………………………..……6

Partie logicielle……………………...…...…….…….6

Analyse et conception………………………………8

Architecture et diagramme……………….…….11

* Architecture logicielle………………….……11
* Diagramme de classe…………………………12

Interfaces……………………...…...…………………13

Mise en production………………………...….….37

Conclusion……………………...…...……………...39

**FIGURES**

**Figure 1 :** Cycle de vie de logiciel en cascade…………....6

**Figure 2 :** Dictionnaire de données. …………...............9

**Figure 3 :** Modèle logique de données (MLD)............10

**Figure 4 :** Architecture Modèle-Vue-Template (MVT)...........................................................................11

**Figure 5 :** Diagramme de classe……………………………..12

**INTERFACES**

**Interface 1** : Page d’authentification à l’espace *Forma Green*………………………………………….*……13*

**Interface 2** : Lien vers le formulaire de création d’un compte.……………………………………*……….14*

**Interface 3** : Première étape d’inscription au *Forma Green*……………………………………*……………..15*

**Interface 4** : Deuxième étape d’inscription au *Forma Green*……………………….……………*…………..16*

**Interface 5** : Troisième étape d’inscription au *Forma Green*……………………………………*…………...16*

**Interface 6** : Validation d’inscription au *Forma Green*………………………………*…………………….……17*

**Interface 7** : Tableau de bord d’association *Forma Green*………………………………*……….…………….18*

**Interface 8** : Tableau de bord d’association *Forma Green (suite)* ………………………….…………*…….18*

**Interface 9** : Gestion des membres de l’association.………………………………..…*………….…………….19*

**Interface 10** : Modal de modification d’un membre d’association. …………………………………….*20*

**Interface 11**: Modal de suppression d’un membre d’association.……………….……………………*….21*

**Interface 12** : Gestion des abonnements.…………………………….……*………………………………………..21*

**Interface 13** : Gestion des activités.……………………………*…….………………………………………………….22*

**Interface 14** : Modal d’ajout d’une activité………………………..…………*……………………*……………...22

**Interface 15** : Modal de modification d’une activité…………..………………………*……………………*….23

**Interface 16** : Modal de suppression d’une activité……………………………………*………………………..23*

**Interface 17** : Gestion de QR Code…………………………………*………………………………………………….24*

**Interface 18** : Affichage de QR Code d’un membre……….…………………………*………………………..25*

**Interface 19 :** Modal d’ajout d’un partenaire…………………………….……*…………………………………..25*

**Interface 20** : Modal de modification d’un partenaire………………………………*…………………………26*

**Interface 21** : Modal de suppression d’un partenaire………………………………*…………..……………...26*

**Interface 22** : Gestion des établissements………………………………*……………………….…………………..27*

**Interface 23** : Modal d’ajout d’un établissement……………………………….…*……………………………..28*

**Interface 24** : Modal de modification d’un établissement………………………………*………..………….28*

**Interface 25** : Modal de suppression d’un établissement………………………………*……..………………29*

**Interface 26** :Liste des membres…………………………………………………………………………..…………30

**Interface 27** : Page d’abonnement d’un membre.………………...……………………………………..……30

**Interface 28** :Liste des activités disponible pour le membre.…………………………...…………………31

**Interface 29** :Modal de détail d’activité participé………………………………………………………………31

**Interface 30** :Liste des partenaires de l’association.……………………………………………………………32

**Interface 31** :Liste des cadeaux proposé offert par l’association.…………………………………………32

**Interface 32** :Liste des établissements………………………………………………………………………………33

**Interface 33** :Badge de membre d’association (Verso)……………………………………………………….33

**Interface 34** :Badge de membre d’association (Recto)……………………………………………………….34

**Interface 35** :La carte des établissements et leurs espaces verts……………………………………………35

**Interface 36** :Exportation et téléchargement de la liste des personnes en format csv…………….36

**Interface 37** : Liste des membres d’association *Forma Green* sous Excel……………………………….36

**Interface 38** : Personnalisation de profil d’un membre/administrateur……………………………..37

**Introduction :**

*Forma Green* est une association qui réintroduit les plantes dans les écoles, universités, écoles de formation, centre de formation…

Cette plateforme permet à chaque membre de bénéficier de remises dans les commerces partenaires contre une participation permettant l’aménagement et l’entretien d’espaces verts ainsi de participer à des activités disponibles dans l’association.

Nos finalités pour ce projet c’est de créer une application web robuste avec une utilisation intuitive pour que l’utilisateur final ait une expérience utilisateur plus améliorée.

C’est pour cela que nous choisissons de décomposer notre projet sur plusieurs parties assurant ses gestions et son développement, tout en commençant par les besoins et les exigences qu’il demande jusqu'à sa réalisation.

Dans ce document, nous vous présenterons en détails les spécifications techniques sur chaque étape et phase qu’ils constituent le projet *Forma Green* en respectant son cahier de charge

**Cycle de vie de logiciel :**

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”.

Graphical user interface

Description automatically generated

**Figure 1 :** Cycle de vie de logiciel en cascade.

**Partie logicielle :**

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

1. **VISUAL STUDIO CODE**

Logo, icon

Description automatically generatedVisual 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.

1. **MySQL**

Logo

Description automatically generatedMySQL 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.

1. **Le Framework DJANGO**

Graphical user interface, website

Description automatically generatedDjango 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 :

* Un langage de gabarits flexible qui permet de générer du HTML, XML ou tout autre format texte ;
* Un contrôleur fourni sous la forme d'un « *remapping* » d'URL à base d'expressions rationnelles ;
* 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.

1. **JAVASCRIPT**

Logo

Description automatically generated

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 cœur du World Wide Web.

1. **BOOTSTRAP**

Icon

Description automatically generated

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 :

Table

Description automatically generated

**Figure 2 :** Dictionnaire de données.

1. 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 :

Diagram

Description automatically generated

**Figure 3 :** Modèle logique de données (MLD).

**Architecture et diagramme :**

1. **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.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 4 :** Architecture Modèle-Vue-Template (MVT).

1. **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 œuvre le système sera une instance des différentes classes qui le compose.

Diagram

Description automatically generatedIci, 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.

**Figure 5 :** Diagramme de classe.

**Interfaces :**

Dans cette partie on va discuter brièvement chaque interface qui compose le projet Forma Green et comment l'utilisateur interagi avec ces interfaces par suite des évènements qui déclenche une opération et des actions spécifiques.

**Authentification/Inscription**

Avant d’accéder au tableau de bord concerné, la première page à laquelle on se redirige directement est la page d’authentification. Dans cette page il suffit d’introduire nos informations de connexion si on a déjà un compte afin d’utiliser la plateforme.

**1) Page d’Authentification :**



Diagram

Description automatically generated

**Interface 1** : Page d’authentification à l’espace *Forma Green*

1. **Page d’inscription :**

Si on veut s’inscrire en tant que membre a ***Forma Green***, il suffit de cliquer sur le lien comme la flèche indique *Créer un compte* afin que nous puissions avoir le formulaire nécessaire de l’inscription à l’espace :

Diagram

Description automatically generated

**Interface 2** : Lien vers le formulaire de création d’un compte.

Après, nous aurons l’occasion de constater qu’après une animation élégante que le formulaire de l’inscription se manifeste pour nous donner la main de s’inscrire en suivant les étapes nécessaires.

**Étape 1 :** Dans cette étape, le formulaire nous indique de choisir si on est une personne physique ou un centre de formation.

Graphical user interface, diagram

Description automatically generatedPar exemple, ici on va choisir qu’on est bien un centre de formation.

**Interface 3** : Première étape d’inscription au *Forma Green*

**Étape 2 :** Après, en cliquant sur le bouton continuez on aura un formulaire permettant d’obtenir nos informations de connexion nécessaires afin qu’on puisse s’authentifier à la plateforme.

Diagram

Description automatically generated

**Interface 4** : Deuxième étape d’inscription au *Forma Green*

**Étape 3 :** En essayant de vérifier nos informations de connexion et après le déclenchement du bouton c*ontinuez*, on aura un autre formulaire mais cela permet de stocker nos informations personnelles à propos de notre centre de formation.

Graphical user interface, diagram

Description automatically generated

**Interface 5** : Troisième étape d’inscription au *Forma Green*

**Fin d’inscription :** Après la validation des champs de formulaire, sans aucune erreur évidemment, on sera redirigé vers la partie de connexion avec un message de confirmation de notre inscription.

Diagram

Description automatically generated

**Interface 6** : Validation d’inscription au *Forma Green*

1. **Application Forma Green :**
2. **Tableau de bord :**

Après, que nous somme identifier avec nos informations de connexion, nous allons être rediriger vers l’application *Forma Green* et plus précisément au tableau de bord qui notre page d’accueil.

**Interface 7** : Tableau de bord d’association *Forma Green*

Graphical user interface, website

Description automatically generatedLa page d'accueil reprend d’une part les informations concernant les membres enregistrés, les partenaires et les établissements inscrits, ainsi que les nombres d’activités effectues dans les espaces verts.

D’autre part, on a un ensemble de photos qui présente les activités déjà réalisées et une brève description concernant la plateforme avec son logo officiel.

Graphical user interface, application, website

Description automatically generated

**Interface 8** : Tableau de bord d’association *Forma Green (suite)*

1. **Partie Administrateur :**

Maintenant, si on se connecte à l’application en tant que qu’administrateur.

On aura la possibilité d’avoir une vue d’ensemble global de l’application et aussi on aura le droit de gérer les membres/adhérents, partenaires, établissements, …

1. **Page liste des membres :**

La liste des membres se divise on deux types essentiels comme le cahier de charge spécifie (Personne/Centre de formation), dans cette liste et en tant qu’administrateur, on a la possibilité de consulter les d’autres page de nos membres suite à la pagination.

**Graphical user interface, table

Description automatically generated**

**Interface 9** : Gestion des membres de l’association.

Ainsi de les modifier en cliquant sur l’icône d’édition, on aura donc un modal qui nous affiche pour nous donner l’occasion de modifier les informations du membres concernés.

Graphical user interface, application

Description automatically generated

**Interface 10** : Modal de modification d’un membre d’association.

Et si on veut supprimer un membre spécifique, en clique sur le bouton supprimer et on aura un message de confirmation de suppression qui nous indique si on veut vraiment supprimer le membre.

Graphical user interface, text, application

Description automatically generated

**Interface 11**: Modal de suppression d’un membre d’association.

1. **Page liste des abonnements :**

**Graphical user interface, application

Description automatically generated**La liste des abonnements permet de voir en clair les abonnements effectués par les membres afin d’être des adhérents, cette liste se constitue comme l’image l’indique, de nom du membre abonné autrement dit l’adhèrent, les nombre de ces abonnements qui a effectué ainsi que la dernière date de son abonnement.

**Interface 12** : Gestion des abonnements.

1. **Page liste des activités :**

**Graphical user interface, application

Description automatically generated**Pour la page des activités, on peut bien consulter toutes les activités décidées et ajouter suite au formulaire d’ajout aussi de les modifier et les supprimer.

**Interface 13** : Gestion des activités.

**Graphical user interface, application

Description automatically generated**

**Interface 14** : Modal d’ajout d’une activité.

**Graphical user interface, application

Description automatically generated**

**Interface 15** : Modal de modification d’une activité.

**Graphical user interface, application

Description automatically generated**

**Interface 16** : Modal de suppression d’une activité

1. **Page liste des QR code :**

Le QR code permet d’identifier et différencier les membres d’association. Dans cette page, on a le droit d’effectuer une recherche d’un membre spécifique suite à un champ multicritère et le type de membre, après la recherche on peut consulter la liste de résultat avec le bouton « afficher QR Code » qui permet de déclencher un modal ayant le QR code du membre désirer afin de le scanner si on a besoin.

**Graphical user interface, application

Description automatically generated**

**Interface 17** : Gestion de QR Code.

**Qr code

Description automatically generated Interface 18** : Affichage de QR Code d’un membre.

1. **Page liste des partenaires :**

**Graphical user interface, application

Description automatically generated**La liste des partenaires nous permet de gérer les partenaires de l’association en les ajoutant via le bouton « ajouter » et on peut également les modifier via le modal de modification et les supprimer après la confirmation du message de suppression.

**Interface 19 :** Modal d’ajout d’un partenaire.

**Graphical user interface, application

Description automatically generated**

**Interface 20** : Modal de modification d’un partenaire.

Graphical user interface, application

Description automatically generated

**Interface 21** : Modal de suppression d’un partenaire.

1. **Page liste des établissements :**

Dans cette interface, on constate qu’on a une liste des établissements qui sont en collaboration avec l’association afin de mettre en place ces activités dans leurs espaces verts, on peut voir également que la liste peut être gérer grâce aux boutons de modification et de suppression aussi, et si on veut bien ajouter un établissement, il suffit de cliquer sur le bouton « ajouter » qui se trouve en haut de la liste.

**Graphical user interface, application

Description automatically generated**

**Interface 22** : Gestion des établissements.

Graphical user interface, application

Description automatically generated

**Interface 23** : Modal d’ajout d’un établissement.

**Interface 24** : Modal de modification d’un établissement.

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Interface 25** : Modal de suppression d’un établissement.

1. **Partie Membre :**

L’interface Membre affiche toutes les informations concernant le membre identifié, que ce soit son profil, ses activités auxquelles il participe et aussi la possibilité de réaliser un abonnement pour devenir un adhérent.

1. **Page liste des membres :**

Le membre ne pourra pas modifier les listes des membres, partenaires et établissements, mais juste visualiser la liste.

Graphical user interface, table

Description automatically generated

**Interface 26** : Liste des membres.

1. **Page mon abonnement :**

Cette page affiche les packs d’abonnements disponibles, ici on a trois types d’abonnements, et l’historique de ses achats qu’il a fait.

**Graphical user interface, website

Description automatically generated**

**Interface 27** : Page d’abonnement d’un membre.

1. **Page d’activités disponible :**

Ici, le membre peut voir les activités disponibles auquel il ne participe pas et la possibilité d’y participer.

**Graphical user interface, application

Description automatically generated**

**Interface 28** : Liste des activités disponible pour le membre.

1. **Page mes activités :**

Dans cette page, le membre peut consulter ces activités auquel il a déjà participé.

**Graphical user interface, application

Description automatically generated**

**Interface 29** : Modal de détail d’activité participé.

1. **Page liste des partenaires :**

Le membre peut consulter la liste des partenaires de l’association.

**Graphical user interface, application

Description automatically generated**

**Interface 30** : Liste des partenaires de l’association.

1. **Page liste des cadeaux :**

Concernant cette page, on a ajouté un seul cadeau, le membre peut acheter et recevoir des stickers avec le logo de l’association.

**Graphical user interface, text, application

Description automatically generated**

**Interface 31** : Liste des cadeaux proposé offert par l’association.

1. **Page liste des établissements :**

Dans cette page, le membre peut consulter la liste des établissements.

Graphical user interface, application

Description automatically generated

**Interface 32** : Liste des établissements.

1. **Badge QR-code :**

Concernant cette partie, on a ajouté une option qui permet de générer un badge Forma Green avec les informations personnelles concernant le membre et son QR-code.

Graphical user interface, application

Description automatically generated

**Interface 33** : Badge de membre d’association (Verso).

Qr code

Description automatically generated

**Interface 34** : Badge de membre d’association (Recto).

Par la suite, on pourra imprimer des badges pour chaque membre, comme ça on aura toutes les informations justes en scannant son QR code qui se trouve sur son badge.

1. **Visualisation des espaces verts :**

Cette page vous permet de visualiser sur une carte, les établissements ayant déjà collaborer avec l’association, c’est-à-dire que vous pouvez cliquer sur un établissement (icône verte) pour avoir une petite fenêtre qui affiche les informations de l’université et zoomer pour voir les zones colorées en vert qui sont les espaces verts implémenter dans l’université.

**Interface 35** : La carte des établissements et leurs espaces verts.

1. **Exporter au format csv :**

**Graphical user interface, application

Description automatically generated**

Les administrateurs de l’association *Forma Green* ainsi les membres peuvent télécharger toutes les listes (membre/partenaires/ établissements, …) en format Excel (.csv) pour qu’ils puissent en cas de besoin, d’imprimer la liste concernée.

Graphical user interface, application

Description automatically generated

**Interface 36** : Exportation et téléchargement de la liste des personnes en format csv.

Graphical user interface, application, table, Excel

Description automatically generated

**Interface 37** : Liste des membres d’association *Forma Green* sous Excel.

1. **Personnalisation de Profil :**

**Graphical user interface, application

Description automatically generated**

**Interface 38** : Personnalisation de profil d’un membre/administrateur.

**Mise en production:**

Pour la partie de la mise en production, on va déployer par la suite l’application web tout d'abord sur Heroku, qui est une plateforme facile à utiliser, permettant le déploiement des applications dans le cloud, elle prend en charge plusieurs langages, parmi eux, Python(Django).

Par la suite, on voudrait maîtriser totalement notre environnement de production, on utilisera un serveur virtuel en hébergement mutualisé.

Voici quelques objectifs à réaliser pour la MEP:

* Se connecter en SSH sur un serveur distant.
* Installer les dépendances sur ce serveur.
* Configurer Nginx et Gunicorn dans notre partie “settings.py” pour l’application Forma Green.
* Utiliser un service de supervision pour Gunicorn.
* Créer un processus d’intégration continue.

**Conclusion :**

Dans cette documentation, nous avons exposé les étapes nécessaires qui aboutissent à la réalisation de ce Projet web avec le Framework Django. Ce projet était une expérience de travail qui nous a permis de bien gérer les tâches, avec chacun leur priorités selon le cahier de charge et aussi la connaissance de certaines notions utiles pour notre expérience de développeur, nous tâcherons d’améliorer l’application web en la rendant plus sécurisé et optimale, et aussi d’intégrer des notions d’apprentissage par machine pour la visualisation des espaces verts, en affichant les universités dont le membre a participé en priorité, avec un système de recommandation des activités les plus proches des membres.