

# Jardins Angevins

**Mieux connaître ce qui pousse dans nos  
jardins**

ACHARD--BONNET César  
MONNIER Alexandre  
TOULON Mathieu

*Octobre 2022 - Janvier 2023*

*Commanditaire :* Conseil de quartier de Belle-Beille, Groupe de travail "Label éco-quartier"

*Professeur référent :* LESAINTE David



# Sommaire

<b>Remerciements</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>I - Naissance de l'application</b>	<b>6</b>
1 - Définition du cadre du projet	6
2 - Recherche des données	6
3 - Structure du projet	7
<b>II - Modèle d'apprentissage et données</b>	<b>8</b>
1 - Récupération des données	8
2 - Redimensionnement et normalisation des images	9
3 - Entraînement du modèle	10
<b>III - Interface mobile</b>	<b>11</b>
1 - Design de l'interface	11
2 - Squelette de l'application	11
3 - Premières fonctionnalités	11
4 - Dernière fonctionnalités	11
<b>IV - Backend</b>	<b>13</b>
1 - Base de données	13
2 - Api et route	13
3 - Tests et documentation	14
<b>V - Gestion du temps</b>	<b>15</b>
<b>Conclusion</b>	<b>16</b>
<b>Annexes</b>	<b>17</b>
Annexe 1: Fiche de projet	17
Annexe 2: Cahier des charges	20
Annexe 3 : Maquette de l'application	22

## Remerciements

Nous tenons avant tout à remercier les différentes personnes qui nous ont permis de réaliser ce projet dans les meilleures conditions

Merci à monsieur David LESANT d'avoir accepté ce sujet de projet et de nous avoir encadré, merci également au service technique du département informatique de l'université d'Angers d'avoir fourni du matériel très utile pour notamment l'apprentissage du système de reconnaissance d'essence. Merci à Amélie THARREAU et Mathias PETITJEAN de nous avoir accueilli et expliqué la volonté du commanditaire. Et enfin merci à la mairie de quartier de belle beille et à son groupe de travail d'avoir proposé ce projet aux étudiants de l'Université.

## Introduction

Dans le cadre de sa démarche de transformation en éco-quartier, le quartier de Belle-Beille, situé dans la ville d'Angers, sollicite l'Université d'Angers et ses étudiants afin de l'aider à réaliser divers objectifs tels que : répertorier, établir et traiter des informations dans le but d'améliorer la connaissance de la flore ainsi que la biodiversité locale, améliorer le cadre de vie du quartier, lutter contre le changement climatique, maintenir une biodiversité stable... Cette démarche s'inscrit dans un contexte de réchauffement climatique, de protection des zones dites "sensibles" mais encore de lutte contre les espèces invasives.

Dans ce contexte ainsi que dans le cadre de notre projet annuel, nous, trois étudiants de Master 2 Informatique au parcours Intelligence Artificielle, avons été intéressés par cette démarche et avons accepté de travailler sur le développement d'un projet d'application mobile.

Cette application mobile a pour but d'aider à la reconnaissance ainsi qu'à la réalisation d'un répertoire des espèces végétales locales selon leur nature et leur position afin de suivre leur évolution dans le temps. Cette application pourra être améliorée avec diverses fonctionnalités supplémentaires.

# I - Naissance de l'application

## 1 - Définition du cadre du projet

Suite à une première réunion avec le client nous avons pu affiner la compréhension de leurs besoins que nous avons pu formaliser en proposant un cahier des charges dans le but de nous définir un cadre de travail clair et une liste d'objectifs à atteindre. Une fois ce cahier des charges rédigé, nous avons pu le renvoyer au client pour qu'il puisse le valider.

Ce projet est donc réalisé et suivi par quatre acteurs à savoir les étudiants qui réalisent l'application, encadrés par un enseignant, M. David LESANT, en second acteur. Le commanditaire du projet qui supervise notre projet mais aussi d'autres initiatives auprès d'autres filières et d'autres étudiants dans le même but de suivre, répertorier et maîtriser la faune et la flore locales, représenté par Amélie THARREAU de la direction du développement des associations et des quartiers. Et enfin le quatrième acteur, un représentant de la population locale Mathias PETITJEAN, lui-même enseignant à l'École Supérieure des Agricultures à Angers apportant un regard plus aguerri sur les connaissances autour du végétale.

## 2 - Recherche des données

Une des premières étapes a été de s'interroger sur l'existence d'une base de données correspondant à nos besoins : une base contenant l'ensemble des essences végétales de la région angevine. En effet, beaucoup de bases de données existent sur internet, mais quand il s'agit d'en trouver une spécifique à une région, ou même une ville, cela reste beaucoup plus compliqué.

De plus, aucune base à notre échelle n'utilise de format similaire entre elle ou au sein même de leurs données, elle n'ont bien souvent pas non plus de manière pratique de récupérer toutes leurs informations, en effet il est bien souvent plus simple de chercher une espèce et d'obtenir toutes les informations souhaitées que d'obtenir une liste en filtrant par des caractéristiques ce qui rend la récupération d'information difficile.

Nous avons donc recherché toute base de données ou répertoire pouvant, de près ou de loin, correspondre à nos attentes en consultant les sites des différentes sources telles que des jardins botaniques, des arboretums ou autres sources dans la région, mais nous n'avons pas réussi à mettre la main sur ce que nous cherchions exactement.

A défaut de cela, nous avons choisi de croiser différentes bases de données correspondant à la région Pays de la Loire ainsi que la base de données Tela

Botanica, un réseau collaboratif de botanistes francophones comme source principale.

Cette solution nous a permis d'obtenir une base de données élargie de ce que nous recherchions. Cependant cela a demandé un travail autour du recoupement et du traitement des informations divergentes ainsi qu'une gestion des informations manquantes par moment.

### 3 - Structure du projet

Avant de nous lancer dans la création du projet nous avons décidé de segmenter le développement en plusieurs parties :

#### **A - IA et données :**

1. Récupération des données via des scripts Python
2. Création d'un modèle de reconnaissance de plantes avec keras

#### **B - Interface utilisateur :**

1. Conception d'une interface utilisateur mobile sur Figma
2. Développement de l'application mobile avec React-Native

#### **C - Backend :**

1. Stockage des données avec Cassandra
2. Serveur web en python avec Flask

Cette structure nous permet d'avoir un backend en Python et d'utiliser un front en JavaScript.

## II - Modèle d'apprentissage et données

### 1 - Récupération des données

Dans un premier temps nous devons trouver les espèces angevines pour spécialiser le modèle d'apprentissage sur ces espèces. Grâce à un site d'inventaire des espèces françaises, Tela Botanica, nous avons sélectionné les espèces inventoriées sur Angers. Ces données sont disponibles sous la forme d'un fichier CSV.

Pour entraîner notre modèle, nous avons aussi besoin d'un ensemble conséquent d'exemples, c'est-à-dire d'images, de photos avec les espèces présentent sur ces dernières. Tela Botanica rassemble de nombreuses images pour chaque espèce, cependant nous n'avons pas découvert sur leur site web la possibilité de télécharger des images pour un ensemble d'espèces ni même d'API permettant de le faire.

Nous avons donc créé un script de scraping qui, pour chaque espèce présente dans le fichier CSV, effectue les actions suivantes :

1. Charger la page HTML spécifique à l'espèce
2. Trouver toutes les images disponibles sur la page
3. Les télécharger localement à partir de leur lien
4. Si l'image est erronée, on la supprime

Cependant nous avons découvert trop tard que Tela Botanica avait une gestion particulière des formats d'images. Chaque lien pour télécharger une image est composé comme suit : [https://api.tela-botanica.org/img:\[identifiant\]\[format\].jpg](https://api.tela-botanica.org/img:[identifiant][format].jpg)

Nous nous sommes rendus compte qu'avec la version précédente, nous téléchargeions des images en doublon : certaines avaient le même identifiant mais pas le même format, de plus certaines images présentes étaient uniquement présentes dans un trop petit format (de l'ordre de 32x32) ce qui peut réduire les performances de l'apprentissage. Tela Botanica dispose de plusieurs formats d'images : CRX2S, CRXS, CXS, CS, CRS, XS, S, M, L, CRL, XL, X2L, X3L, O.

Nous privilégions le format O car il s'agit de la photo originale (sans redimensionnement).

La seconde version du script de scraping effectue pour chaque espèce les actions suivantes :

- 1 - Charger la page HTML spécifique à l'espèce
- 2 - Trouver tous les identifiants d'image présents sur la page



- 3 - Créer une liste d'identifiants uniques (pas de doublon).
- 4 - Pour chaque identifiant construire le lien en fixant le format.
- 5 - Les télécharger localement à partir du lien
- 6 - Si l'image est erronée, on la supprime

## 2 - Redimensionnement et normalisation des images

Pour pouvoir utiliser les images pour l'entraînement grâce à un réseau de neurones, nous devons redimensionner les images pour qu'elles soient toutes de la même taille. Nous avons testé plusieurs méthodes :

- Redimensionnement
- Recadrement au centre de l'image (on suppose que l'information principale est au centre)
- Redimensionnement ainsi qu'ajout de pixels noirs autour pour garder la ratio longueur / largeur
- Génération de trois images à partir d'une : recadrement au milieu, à droite du milieu et à gauche du milieu.

Le dernier algorithme est celui que nous avons retenu car il est le plus efficace. Cet algorithme fonctionne comme suit :

- **Si** : l'image est carrée alors on la redimensionne en 256x256 puis on génère l'image
- **Sinon**: On redimensionne le plus grand côté à 256 et le plus petit en gardant le ratio de l'image.
  - On calcule la position du centre de l'image redimensionnée
  - **On génère** l'image recadrée 256x256 correspondant au centre de l'image rectangulaire (en noir)
  - **On génère** l'image recadrée 256x256 comprenant la partie plus à droite de l'image et qui contient le centre de l'image (en rouge).
  - **On génère** l'image recadrée 256x256 comprenant la partie plus à gauche de l'image et qui contient le centre de l'image (en bleu).

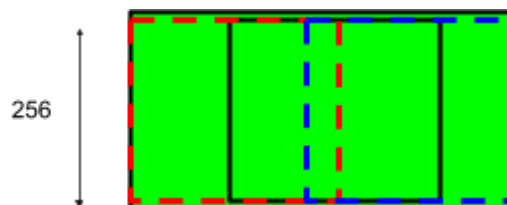


Figure 1 : Cas d'une image plus large que longue

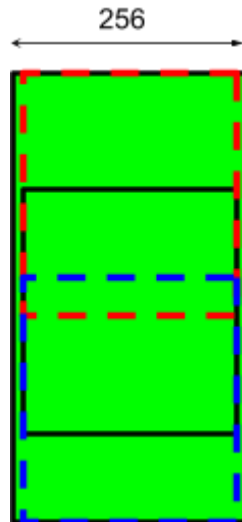


Figure 2 : Cas d'une image plus longue que large

Nous pouvons voir dans les deux figures ci-dessus comment fonctionne l'algorithme après redimensionnement. Chaque cadre représente une nouvelle image qui va être générée.

### 3 - Entraînement du modèle

Maintenant que nous avons des données utilisables il suffit de lancer l'entraînement. Nous utilisons la librairie autokeras, un système de machine learning clé en main basé sur keras.

Nous avons rencontré plusieurs erreurs pendant l'entraînement. C'est pourquoi nous avons créé un script de debug qui permet de savoir quelles sont les espèces pour lesquelles des images posaient problème. Ce script nous a permis de corriger de nombreux problèmes liés au scraping (certaines images étaient vides) ainsi qu'au redimensionnement (cas spécifique non pris en compte).

## III - Interface mobile

### 1 - Design de l'interface

Avant de débiter le développement de l'interface et de l'application mobile en elle même nous devions créer une maquette, nous avons donc tout d'abord défini une charte graphique avec police d'écriture et couleurs de l'application. Puis nous avons réfléchi aux différentes possibilités que devait offrir l'application pour enfin concevoir un prototype de maquette avec chacune des interfaces. Maquette qui a été amenée à évoluer à de multiples reprises avant sa version finale. Pour réaliser cette maquette nous avons utilisé Figma.

### 2 - Squelette de l'application

Dans un premier temps nous avons réalisé un squelette de l'application qui reprend donc la maquette précédemment conçue, et Figma permet de générer du code HTML et CSS à partir du design ce qui a permis de générer la version pour React Native plus rapidement vu que ce framework est volontairement très proche des technologies du web. Ce squelette ne comportait aucune fonctionnalité en soit mais permettait juste une navigation.

### 3 - Premières fonctionnalités

Une fois le squelette de l'application mis en place, nous avons pu mettre en place chacune des fonctionnalités en commençant tout d'abord par les plus simples avec notamment le chargement des données des différentes espèces sur le menu et les pages de wiki.

Toute cette base a été mise en œuvre en utilisant une architecture service - vue - composants - assets ce qui permet de rendre l'intégration de nouvelles fonctionnalités et la mise à jour de celles déjà présentes plus facile. Ainsi qu'une réutilisation maximale de certains éléments pour rester DRY.

### 4 - Dernière fonctionnalités

Certaines fonctionnalités et éléments d'interface ont été plus complexes à développer, en effet deux parties se sont révélées relativement complexes.

Premièrement la carte, sous le framework fournit il nous a été impossible d'utiliser une carte open source telle que openstreetmap sans devoir refaire toute

l'implémentation car React Native reste relativement jeune. Et cette relative jeunesse du framework et de ces librairies entraîne aussi des problèmes que nous avons rencontré qui sont encore aujourd'hui non résolus dans les dernières versions. Un exemple concret est par exemple dans certaines conditions particulières que nous réunissons dans l'application, les marqueurs sur une carte google maps se téléportent aléatoirement.

Enfin une seconde difficulté a été le traitement de l'image, en effet nous avons souhaité avoir un backend strict qui ne refait pas de traitement des images fournies il s'agit donc au client de faire le traitement, de fournir au format attendu l'image. Cependant, dans un framework de haut niveau comme celui que nous utilisons, il n'est pas concevable de refaire nous même un changement de format, un redimensionnement et un redécoupage de l'image, il faut donc chercher parmi les librairies communautaires lequel nous permet d'obtenir le résultat souhaité. Ce qui n'a pas été une tâche facile, en effet, après avoir trouver plusieurs librairies certaines comportent soit leurs failles de sécurité connues, soit ne sont plus mises à jour ou sont dépréciées ou encore certaines comportent des erreurs connues qui, par malchance, concernent aussi notre application. Cependant l'avantage du côté communautaire est aussi d'avoir d'autres personnes ayant eu les mêmes problèmes avant nous et qui apportent leur version ce qui nous a dans ce dernier cas permis d'obtenir une solution.

## IV - Backend

### 1 - Base de données

Nous avons fait le choix d'utiliser Cassandra un SGBD souvent utilisé pour stocker un gros volume de données comme Discord a pu l'utiliser dans le passé pour stocker toutes les données des utilisateurs et de leurs nombreux messages mais aussi couramment utilisée par de nombreuses entreprises et startups dans le domaine de l'intelligence artificiel. Son point négatif majeur aujourd'hui est sur la suppression de données en quantité qui si elle est trop importante pose problème mais dans notre cas il n'est pas prévu de supprimer des données, pour le moment, et encore moins de manière massive.

La base de données est découpée en deux tables :

- **UserInput** : Pour enregistrer les images des utilisateurs, localisation de l'image, la prédiction du modèle ainsi que le moment où elle a été prise.
- **Species** : Pour enregistrer des données fiables sur les espèces angevines récupérées via Tela Botanica telles que le numéro nomenclatural, leur nom, leur nom scientifique, l'image de référence et les autres images associées ainsi que des statistiques sur l'espèce (consommation d'eau et lumière, toxicité)

La base de données est connectée à la gestion centrale du backend par la librairie cqlengine en python, pour conserver un backend en full python.

### 2 - Api et route

Il existe au total cinq routes que nous allons détailler dans cette section. Ces routes permettent à l'application de dialoguer avec la base de données et le modèle d'apprentissage.

Voici le détails des différentes routes :

- **query** : à partir de la position gps et d'une image fournie en base64. Elle nous permet d'obtenir la prédiction de notre modèle, puis d'enregistrer les informations telles que l'image, la prédiction et la date dans la base de données. Et enfin elle renvoie à la meilleure prédiction.

- **statistics** : permet de récupérer diverses informations telles que le nombre de contributions, le nombre de téléchargement de l'application etc.
- **species/wiki** : retourne les informations sur une espèce donnée à partir du numéro nomenclatural telles que le nom scientifique, l'image de référence ainsi que des statistiques sur l'espèce (consommation d'eau besoin en luminosité, toxicité).
- **species/list** : retourne l'ensemble des espèces disponibles avec le numéro nomenclatural, le nom et l'image de référence. La liste retournée est paginée pour des raisons d'optimisation.
- **map** : à partir d'une position gps ainsi que d'un delta en longitude et latitude, cette route retourne tous les points enregistrés sur la surface demandée. De plus, des paramètres supplémentaires tels qu'une année ou un numéro d'espèce peuvent être fournis pour filtrer les points retournés. Enfin cette route possède en plus une sécurité configurable pour éviter qu'un utilisateur ne demande une surface trop grande à retourner.

La principale difficulté que nous avons rencontré dans la mise en place de ces routes est le dialogue avec la base de données Cassandra qui utilise un langage particulier appelé cql. Les requêtes effectuées ne sont pas toutes optimisées car la sélection des n-uplet ne se fait pas directement grâce au sgbd mais à l'aide de traitement en python. Il serait judicieux à l'avenir de faire directement les opérations avec le sgbd cela rendrait les requêtes plus rapides.

### 3 - Tests et documentation

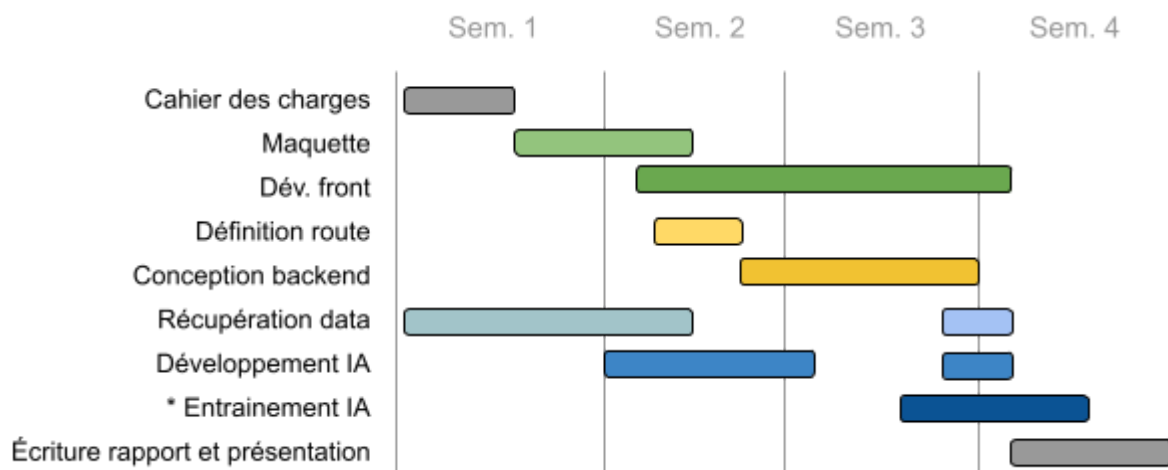
Chacune des routes évoquées dans la précédente partie est testée en utilisant directement le système de tests inclus dans les modules par défaut de python. Ainsi chaque route est testée individuellement aussi bien sur des valeurs attendues que sur des erreurs.

Chacune des routes et de ces comportements, ainsi que les détails techniques relatifs à chacune d'entre elles, tels que le format d'image fourni par l'API et celle en entrée qui est attendue, les différents code de retour et leurs motifs.

Aussi ce projet ayant pour but d'être une preuve de concept la documentation contient non seulement les informations évoquées précédemment mais aussi des informations utiles pour les développeurs qui souhaitent reprendre le code à la suite

de notre projet avec des informations sur comment redéployer , réinstaller le projet et les spécificités de ce dernier.

## V - Gestion du temps



La gestion du temps a été une réelle problématique durant ce projet, nous avons de base posé des délais qui ont été presque parfaitement respectés. Cependant la problématique de la proposition de remplacement d'une espèce n'a pas été traitée à la fois pour une question de délais mais aussi à cause d'un manque d'information sur cette notion de substitution.

Vous retrouverez sur la figure ci-dessus une indication de la répartition du temps que nous avons alloué au projet, tout en sachant qu'ici les périodes sont indiquées comme continues ce qui n'a pas été le cas, ce qui impacte notamment la partie sur l'entraînement de l'IA qui a bénéficié de la période de Noël.

## Conclusion

Ce projet est enrichissant car il nous a permis d'approfondir nos connaissances dans plusieurs disciplines : Machine learning, Développement Back-end, Développement mobile et base de données NoSQL.

Nous avons réussi à construire les fondations, le projet est fonctionnel, documenté et testé. Il y'a cependant certains axes d'amélioration :

- La reconnaissance d'image pourrait être améliorée en utilisant une structure adaptée au problème ( cf. transformer).
- Sécurisation de l'API.
- Système de vérification des données utilisateur (afin d'éviter d'avoir des données erronées ).
- Application non testée sur IOS
- Problématique de la proposition de remplacement d'espèce



## Annexes

### Annexe 1: Fiche de projet

# FICHE PROJET

## “Mieux connaître ce qui pousse dans nos jardins ”

*Commanditaire : Conseil de quartier de Belle-Beille  
Groupe de travail “label éco-quartier”*

### Contexte de la commande

---

*Qu’est ce que le groupe de travail “label éco-quartier” ?*

Le conseil de quartier est une instance consultative et participative présente sur l’ensemble de quartiers de la ville d’Angers. Chaque conseil de quartier est composé de 30 citoyens bénévoles, appelés des “conseillers de quartier”, qui agissent sur leur quartier de résidence pour des mandats de 2 ans.

Les conseils de quartier ont été établis dans le but de récolter l’avis des habitants sur les différents projets élaborés à l’échelle de la ville ou des quartiers et d’accompagner les décisions municipales. Les conseillers de quartiers sont ainsi invités à participer aux différentes phases d’élaboration des projets en transmettant leurs avis et leurs idées. Ce sont de véritables forces de proposition et d’évaluation.

A Belle-Beille, le conseil de quartier délibère sur des sujets réguliers mais a également choisi de porter son regard sur 3 enjeux qui marquent le quartier : le décrochage scolaire, la visibilité des services existants sur le quartier pour les nouveaux habitants, la labellisation éco-quartier. Chaque thématique est suivie par un groupe de travail composé de différents conseillers de quartier. C’est dans ce cadre que le groupe de travail “label éco-quartier” a été constitué. Ce dernier s’intéresse à la démarche de labellisation dans laquelle est entrée Belle-Beille et souhaite initier des projets en lien avec les objectifs auxquels doit répondre le quartier. Pour ce faire, le conseil de quartier est notamment accompagné par le Pôle Territorial Belle-Beille - Lac de Maine.

### *Pourquoi faire cette commande aux étudiants ?*

Dans ce cadre, le groupe de travail souhaite faire appel aux étudiants afin de les aider dans leurs réflexions. En effet, de véritables besoins en termes d'acculturation à différentes thématiques liées à la notion d'éco-quartier et à la connaissance du quartier en lui-même ont été détectés au sein du conseil. Les travaux étudiants initiés seraient alors de véritables ressources pour amorcer des projets et diagnostiquer les enjeux et objectifs prioritaires d'action sur le quartier. Derrière ce partenariat, c'est aussi toute une dynamique de mise en lien entre étudiants et habitants du quartier qu'ambitionne d'impulser le conseil de quartier.

### **Objet de la commande**

---

*Sujet : Mieux connaître ce qui pousse dans nos jardins*

*Les constats à l'origine de ce sujet :*

- La démarche Eco-quartier permet d'envisager à l'échelle du quartier des actions pour améliorer le cadre de vie, lutter contre le changement climatique et maintenir une biodiversité. Depuis l'urbanisation du plateau de Belle-Beille, une végétalisation de toutes sortes a été implantée ou s'est développée spontanément.
- Cette végétation est hétérogène du fait de la diversité des jardiniers (publics et privés)
- Initiative "Arbre remarquable" : questionnaire déjà existant.

*Les objectifs de cette commande :*

- Un besoin de connaissance et d'amélioration des pratiques au service du quartier.
- Travailler sur la connaissance des essences du quartier par les habitants
- Recueillir les besoins des jardiniers pour entretenir les végétaux (les néo-jardiniers qui souhaitent avoir un soutien par un jardinier) - Recueillir des savoirs-faire potentiels des jardiniers (privés)
- Aider les jardiniers à choisir des essences ou des pratiques qui peuvent répondre aux besoins (biodiversité - faune - flore - lutte contre changement climatique (végétation mieux adaptée à l'augmentation des températures, restriction de l'eau,...))
- Favoriser la lutte contre les essences invasives

- Possibilité de faire travailler sur ce sujet différentes facultés.

## Livrable

---

- Charte de bonnes pratiques culturelles (essences à planter et entretien)
- Recensement pour un catalogue des essences adaptées ou à mettre en avant auprès des jardiniers
- Application mobile de reconnaissance des essences et conseils d'entretiens, en lien avec la démarche "Territoire intelligent". Exemple : <https://www.google.com/url?q=https://monjardinmamaison.maison-travaux.fr/mon-jardin-ma-maison/actualites-jardin/applications-identifier-plantes-grace-a-votre-smartphone-281351.html%23item%3D1&sa=D&source=docs&ust=1652720960025212&usg=AOvVaw3eBXuBZ9firtjY6Es016qU>

## Modalités d'organisation

---

### *Éléments d'organisation :*

- Date/période d'intervention envisagée : à déterminer avec les équipes pédagogiques et suivant les priorisations futures.
- Proposition de rencontres et temps d'échanges avec les étudiants ?

### *Contacts des personnes ressources :*

- Mathias Petitjean (conseillers de quartier) : [mathias.petitjean@gmail.com](mailto:mathias.petitjean@gmail.com)
- Didier Margot (conseillers de quartier) : [didier.margot.dm@gmail.com](mailto:didier.margot.dm@gmail.com)
- Léonie Decron (chargée de projet Campus-Quartier, Pôle territorial) : [leonie.decron@ville.angers.fr](mailto:leonie.decron@ville.angers.fr)

## Annexe 2: Cahier des charges

# Cahier des charges Application mobile Mieux connaître ce qui pousse dans nos jardins

18 octobre 2022

Mathieu TOULON  
Alexandre MONNIER  
César ACHARD--BONNET

mathieu.toulon@free.fr  
alexandremonnier012@gmail.com  
cesar.achardbonnet@gmail.com

### Résumé

Dans le cadre de sa démarche de transformation en écoquartier, le quartier de Belle-Beille sollicite l'Université d'Angers afin de l'aider à répertorier, établir et traiter des connaissances dans le but d'améliorer la faune, la flore ainsi que la biodiversité locale. Cette démarche s'inscrit dans un contexte de réchauffement climatique, de protection des zones dites "sensibles" mais encore de lutte contre les espèces invasives.

Dans ce contexte et dans le cadre de leur projet étudiant annuel, trois étudiants de Master 2 Informatique parcours Intelligence Artificielle sont mandatés sur un projet d'application mobile visant à aider la reconnaissance ainsi que la réalisation d'un répertoire des espèces végétales locales selon leur nature et leur position afin de suivre leur évolution dans le temps. Cette application fournira des propositions de substitution d'espèces végétales non désirées en se basant sur différents critères tels que la nuisibilité le caractère invasif de l'espèce par exemple.

## 1 Fonctionnalités

Les fonctionnalités à développer pour ce projet sont :

1. Reconnaissance d'essence :
  - Récupération des données open-source et libres d'utilisation
  - Entraînement d'un modèle grâce au machine learning
2. Inventaire des essences :
  - Horodatage
  - Géo-localisation
  - Évolution spatio-temporelle des essences (carte)
3. Proposition d'une essence alternative : On détermine à partir d'une essence donnée une ou plusieurs espèce-s alternative-s en considérant certains critères :
  - essence semblable non invasive
  - essence semblable avec consommation plus faible (en eau, nutriments etc.)
  - essence semblable résiliente au changement climatique.

## 2 Données et informations

Nous distinguerons deux types de données :

- Les données de confiance : Ce sont des données récoltées à partir de de bases de données **open-source** et **libres d'utilisation**.
- Les données utilisateurs : lorsqu'un utilisateur prend une photo avec l'application, celle-ci est horodatée et géo-localisée. Ces données ne sont pas fiables c'est pourquoi la distinction et ici faite. Ces données seront tout de même utiles dans notre système d'apprentissage.

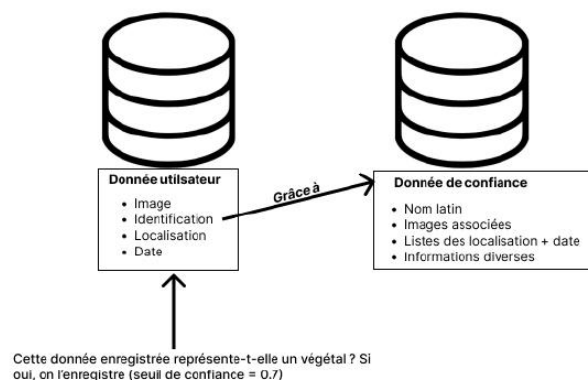


FIGURE 1 – Schéma de l'organisation des données

Lorsque un utilisateur prend une photo, les données sont envoyées au serveur qui détermine d'abord si la donnée caractérise un végétal, auquel cas on se basera sur les données dites "de confiance" pour déterminer l'essence photographiée grâce à l'aide de notre modèle de machine learning.



## Annexe 3 : Maquette de l'application

### Jardins Angevins

