

Spécification des exigences

Turing Gallery



06 novembre 2019

ESIF Digital Corporation

Version 1.0

1. Introduction

1.1 Objet

L’objet de ce document est de décrire l’environnement, les fonctions et les contraintes, le type des utilisateurs finaux ainsi que les exigences fonctionnelles et non fonctionnelles, du système de générateur de catalogue automatique « Turing Gallery » assisté par Machine Learning. Ce document est destiné principalement aux développeurs du logiciel mais aussi à toute autre personne voulant découvrir en détail le fonctionnement du logiciel.

1.2 Portée

Le logiciel « Turing Gallery » permettra aux photographes et aux artistes de publier leurs photos et œuvres sur l’application, une intelligence artificielle s’occupera d’organiser ces images dans un ordre qui s’aligne sur les préférences de l’utilisateur, puis un catalogue sera généré à partir du résultat final. Le système est prévu pour être utilisé par des professionnels dans l'événementiel, il faut donc s’attendre à une utilisation pouvant être intensive avec un grand nombre de connexions, et également un espace de stockage assez large pour supporter l’utilisation de ce dernier.

1.3 Glossaire

Le tableau ci-dessous expliquera les différents vocabulaires techniques pouvant être susceptible de freiner la compréhension de ce document :

|  |  |
| --- | --- |
| Terme ou acronyme | Définition |
| Java | Langage informatique très populaire, doté d’un écosystème très complet et important. |
| Framework | Ensemble d’outils et de composant logiciel organisés conformément à un plan d’architecture |
| Spring | Framework Java pour faciliter le développement d’application web |
| Apache Maven | Gestionnaire de dépendances, très utilisé pour gérer les différentes dépendances pouvant être présente dans une application |
| Git | Outil de gestion de version |
| GitHub | Plateforme d’hébergement d’outil de version, ici git |
| NetBeans | Environnement de Développement Intégré : utilisé pour accélérer le développement d’application |
| Visual Studio Code | Environnement de Développement Intégré : utilisé pour accélérer le développement d’application |
| Heroku | <https://www.heroku.com/> |
| Cloud | le Cloud consiste à utiliser des [serveurs informatiques](https://fr.wikipedia.org/wiki/Serveur_informatique) distants par l'intermédiaire d'un réseau, généralement [Internet](https://fr.wikipedia.org/wiki/Internet_Protocol), pour stocker des données ou les exploiter. |
| Scalabilité | Désigne la capacité d'un produit à s'adapter à un changement d'ordre de grandeur de la demande (*montée en charge*), en particulier sa capacité à maintenir ses fonctionnalités et ses performances en cas de forte demande |
| Déploiement | Le déploiement de logiciels regroupe toutes les activités qui rendent un système logiciel utilisable et accessible aux utilisateurs finaux |
| MVC | Modèle-vue-contrôleur ou MVC est un motif d'[architecture logicielle](https://fr.wikipedia.org/wiki/Architecture_logicielle) |

*Figure 1 : définitions et acronymes*

1.4 Vue d’ensemble

----

2. Description générale

2.1 Interfaces du logiciel

2.1.1 Interfaces système

----

2.1.2 Interfaces utilisateurs

----

2.1.3 Interfaces matérielles

----

2.1.4 Interfaces logicielles

----

2.1.5 Interfaces de communication

----

2.1.6 Contraintes de mémoires

Pour l’hébergement et le déploiement de l’application Web, nous avons opté pour une Plateforme Cloud chez Heroku, c’est l’une des meilleures solutions pour héberger l’application Web car :

* Supporte les projets développés en langage Java et propose un service adapté aux projets développés avec le Framework Spring.
* La prise en charge du côté serveur est gérée par Heroku, donc les développeurs peuvent se concentrer sur le code sans avoir besoin de l’adapter.
* Les services et les ressources sont extensibles et scalables.
* La communication avec le serveur est sécurisée.
* Le déploiement de l’application est automatisé via Git.

2.2 Fonctions du logiciel

----

2.3 Caractéristiques des utilisateurs

----

2.4 Contraintes

----

2.5 Hypothèses et dépendances

----

3. Exigences spécifiques

3.1 Exigences fonctionnelles

----

3.2 Exigences non fonctionnelles

----