



# Hidden Words – Projet final

Dossier de conception fonctionnelle

Version 1.1

**Auteur**

Axel Mesnard  
Développeur Python



## Table des matières

<b>1 - Versions .....</b>	<b>3</b>
<b>2 - Introduction .....</b>	<b>4</b>
2.1 - Objet du document .....	4
2.2 - Références.....	4
2.3 - Besoin du client .....	4
2.3.1 - Contexte.....	4
2.3.2 - Enjeux et Objectifs.....	4
2.4 - Les principes de fonctionnement.....	4
Les acteurs.....	5
2.5 - Les cas d'utilisation généraux.....	6
<b>L'Application Web .....</b>	<b>7</b>
3.1 - Les acteurs .....	7
3.2 - Les cas d'utilisation.....	7
3.2.1 - Package Front Office.....	7
3.2.1.1 - Cas n°1.....	7
<b>4 - Glossaire.....</b>	<b>9</b>



# 1 - VERSIONS

Auteur	Date	Description	Version
Axel	14/01/2022	Création du document	1.0
Axel	19/01/2022	Finalisation du document	1.1



## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application Hidden Words, un générateur de mots cachés/mêlés.

L'objectif du document ici est de décrire les détails, les fonctionnalités de l'application Hidden Words.

### 2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **DCT** : Dossier de conception technique de l'application.
2. **DE** : Dossier d'exploitation.

### 2.3 - Besoin du client

#### 2.3.1 - Contexte

Faisant partie d'un groupe qui est fan des mots fléchés, mots cachés/mêlés, etc, je me suis rendu compte qu'il pouvait être intéressant de créer un générateur de mots cachés/mêlés afin qu'ils n'aient plus à en acheter. Après m'être rendu compte qu'un générateur de mots cachés/mêlés, afin que nous n'ayons plus à en acheter en commerce.

#### 2.3.2 - Enjeux et Objectifs

Cela permettra à un utilisateur de simplement se connecter à mon site web afin de télécharger une grille en PDF générer aléatoirement.

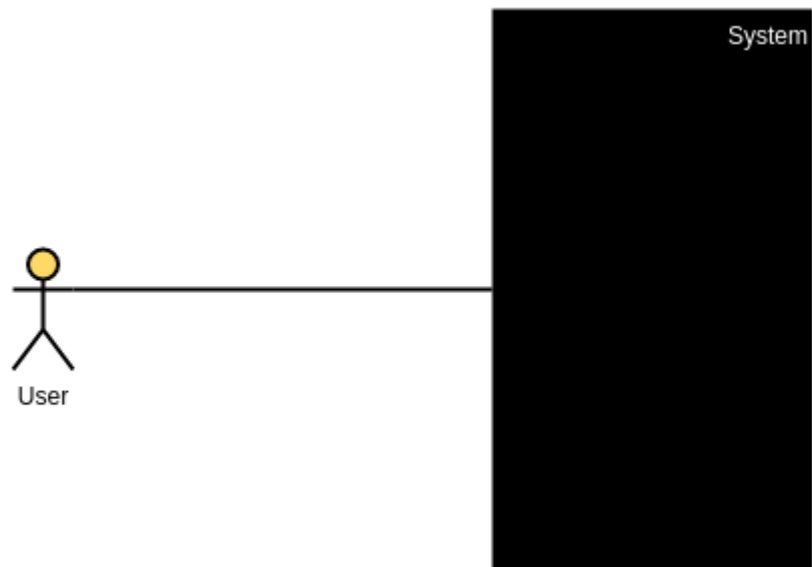
### 2.4 - Les principes de fonctionnement

L'application Hidden Words sera une application web, plus communément appelé une web-app. Le langage principal utilisé sera le Python pour l'application en elle-même. Le site sera en HTML/CSS/JS.



## 2.5 - Les acteurs

### Diagramme UML de contexte

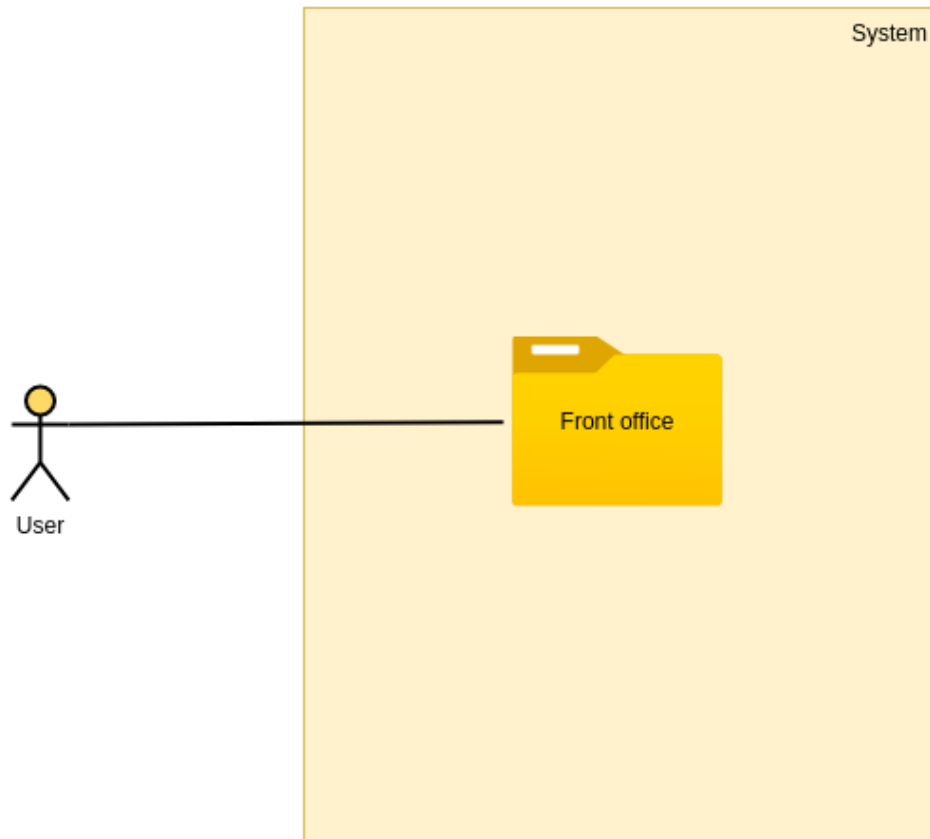


Voici les différents acteurs qui interagiront avec le futur système. Nous pouvons voir à gauche l'acteur « physique ». Ce système est très simple car l'application est elle-même simple en termes de parcours utilisateurs et des personnes qui vont interagir avec elle.



## 2.6 - Les cas d'utilisation généraux

### **Diagramme UML des cas d'utilisation généraux.**



En rentrant plus en détail dans le système, nous pouvons voir que nous aurons un package « Front Office » :

- Le « Front Office », c'est donc la partie visible du site web par le user. On y accède par l'adresse internet classique et on peut y télécharger directement le fichier PDF.



## 3 - L'APPLICATION WEB

Nous pouvons ici nous référer au contexte et aux enjeux et objectifs énumérés dans les sections [2.3.1](#) et [2.3.2](#).

### 3.1 - Les acteurs

Ce sont les mêmes que ce ceux qui ont été présentés lors de la description générale de la solution au chapitre [3.2](#), via le diagramme UML de contexte.

### 3.2 - Les cas d'utilisation

#### 3.2.1 - Package Front Office

##### 3.2.1.1 - Cas n°1

Nom : L'utilisateur se connecte au site et télécharge un PDF généré aléatoirement

Acteur(s) : Utilisateur

Description : L'utilisateur vas se connecter au site, sélectionner le nombre de mots et la taille de la grille qu'il souhaite. Puis télécharger le fichier en cliquant sur le bouton.

Auteur : Axel Mesnard

Date(s) : 15/11/2021 (première rédaction)

Préconditions : Aucunes

#### DESCRIPTION

Le scénario nominal :

1. L'utilisateur rentre l'adresse du site pour y accéder.
2. Le système affiche la page d'accueil unique qui présente le générateur avec un peu de texte. Deux options sont disponibles pour personnaliser le PDF, à savoir la taille de la grille ainsi que le nombres de mots que l'on peut mettre à l'intérieur.
3. L'utilisateur choisi donc la taille de la grille ainsi que le nombre de mos qu'il souhaite avoir à l'intérieur.
4. L'utilisateur clique sur le bouton "générer un PDF".
5. Le système prend envoie la commande à AWS qui via plusieurs services renvoyer le lien de téléchargement du PDF générer aléatoirement.
6. L'utilisateur peut maintenant choisir d'ouvrir le PDF ou de l'enregistrer, avec peut-être des options supplémentaires en fonction de son navigateur.



## COMPLEMENTS

### Ergonomie

Le site web utilisera un template bootstrap modifié pour faire un simple onepage.

### Performance attendue

Le site web sera responsive, cependant le temps d'attente pour générer le PDF pourra prendre quelques secondes, l'appel à la base de données est la partie qui peut prendre un peu de temps car nous restons ici sur le mode moins performant de DYNAMODB, car sinon il faut payer plus chère. Il faut donc prendre en compte que le temps de réponse peut être drastiquement réduit si l'on met plus d'argent, ce n'est pas l'idée pour ce projet.





## 4 - GLOSSAIRE

<b>UML</b>	Le Langage de Modélisation Unifié, de l'anglais <i>Unified Modeling Language</i> (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.
<b>Python</b>	Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels.