

## Énoncé de projet de fin d'année Outil TOOTLIST

### But initial

Afin de résumer (et valider) vos acquis en cette 3<sup>e</sup> année, un projet à long terme vous est proposé en ce début d'année. Vous réaliserez celui-ci en groupes de **4 personnes maximum**. Une partie gestion de projet (contrôlée par Hervé BARBOT) viendra évaluer documentation et suivi, un bilan régulier sera établi afin de juger de votre avancée. Vous effectuerez un rendu final, et une note sera émise lors d'une présentation (devant jury extérieur, avec démonstration technique et document vidéo-projeté).

L'esprit de ce projet est de vous fournir une demande non formalisée : juste un peu plus complète qu'un appel d'offre, et moins structurée qu'un cahier des charges fonctionnel. Vous trouverez, d'une part, des informations sur la forme à donner à votre solution logicielle, et d'autre part quelques restrictions architecturales. Dans les grandes lignes, le projet est donc très **libre** : vous choisissez les **technologies** à déployer (celle-ci étant un facteur d'évaluation du rendu final : prise de risque et choix pertinents sont de rigueur !). Vous devrez respecter certaines **contraintes** : celles-ci ont pour but de cadrer un minimum vos choix et de vous obliger à **vous confronter à certains problèmes**.

### De quoi s'agit-il ?

Le projet « **Tootlist** » part d'un constat simple : tout peut être « listé », chaque sujet peut prendre la forme d'une liste. Mais, après avoir été constituée, une liste doit pouvoir être complétée, mise à jour, organisée, triée, hiérarchisée, recherchée, retrouvée, gérée, enrichie...

Le but de l'application « **Tootlist** » est donc de rendre cela possible, de façon simple, ergonomique et sécurisée, en mode connecté ou déconnecté.

En pratique, imaginez une immense base de données connectée directement à votre cerveau, regroupant l'ensemble de vos projets, de vos idées, de vos résolutions, de vos envies, de vos tâches et courses quotidiennes, de vos numéros utiles et importants, de vos amis... mais qui archive également l'ensemble de vos photos, disques, films, livres, sites... préférés, qui répertorie idées de recettes, de cadeaux, de voyages, de lieux à visiter, vous rappelle les fêtes, les anniversaires, les factures à payer, les rendez-vous, les « choses » à ne pas oublier et qui vous permet, enfin, de créer toute nouvelle liste de votre choix.

Bref, un Pense-bête intelligent personnalisable à l'infini, réunissant et organisant dans une même application Todo-list, gestion de tâches, répertoire, mémo, planning, annuaire... consultable de n'importe où à partir d'un accès à Internet ou d'un Smartphone de type Iphone et régulièrement enrichi par la communauté des utilisateurs.

### La solution s'appuiera sur :

1. **Un serveur de bases de données**, regroupant toutes les informations de l'outil
2. **Une application web** tirant parti de cette base, proposant à la fois un front office (l'outil à proprement parler, utilisé par ses visiteurs, après inscription puis authentification.
3. **Une application embarquée ou autre, NON-WEB** (ne s'appuiera pas sur de l'architecture client/serveur pour fonctionner) qui permettra de récupérer ponctuellement, puis d'utiliser en mode déconnecté des données produites par le site (ex : sur mon Iphone, une application me permet de charger ma liste de courses, de l'utiliser, la modifier, même lorsque la connexion est perdue, puis de remonter les modifications sur le site par synchronisation).

## Énoncé de projet de fin d'année Outil TOOTLIST

### LE SITE WEB :

#### La page d'accueil

Celle-ci devra s'inspirer de sites emblématiques de la sortie des concepts web2.0 (Netvibes, iGoogle, ...). C'est-à-dire que la page contiendra un certain nombre de « boîtes », ordonnées ou non, mais quoi qu'il arrive placées par l'utilisateur. Vous devrez offrir la possibilité de faire cette mise en page par Glisser-Déposer.

Vous devrez mettre à disposition de l'utilisateur :

- Une boîte de login, permettant la récupération des préférences utilisateur, et surtout l'accès à la lecture et modification des données de liste de celui-ci.
- Un menu de navigation sur le site : plan du site, contacts, aide, et surtout Edition des listes personnalisées !
- Une zone « fixe », contenant un accès rapide à des modèles de liste, ou aux listes les plus populaires des utilisateurs.
- Une zone variable, avec les boîtes en glisser-déposer, qui contiendra ce que l'utilisateur décide de placer. A la première visite, l'utilisateur devra disposer d'un modèle basique, qu'il modifiera à sa guise.

#### Contraintes techniques :

- Le glisser déposer n'est pas une « option » : il fait partie des obligations
- Pour gérer ces animations, le choix judicieux d'un bon framework et une implémentation rigoureuse de celui-ci auront bien plus de valeur qu'une réécriture de code javascript !
- Les données utilisateur n'ont pas de raison d'être stockées en local : vous pouvez tout placer en base.
- L'animation des menus, ainsi qu'un design clair, mettant en valeur l'ergonomie sont un plus non négligeable.

### Types de liste, catégories et outils de recherche

Le site (et nous le verrons plus tard l'application classique) propose une prestation aussi bien riche en services qu'adaptable à toute forme de besoin. C'est-à-dire ? Et bien ce là signifie que **Tootlist** doit permettre de construire un certain nombre de listes déjà utilisables telles-que (liste de MP3 avec lecteur intégré, liste de photos avec aperçu dynamique au survol ...) mais également de fabriquer des listes souples, sans types de données préconçues, sans structure stéréotypée. L'utilisateur doit pouvoir générer son propre modèle.

Chaque groupe devra implémenter un certain nombre de listes, piochées dans un tableau, présenté ci-dessous :

- **3 listes du lot 1** : ce lot contient des listes simples, offrant naturellement très peu de service
- **2 listes du lot 2** : des types de listes avec des services conséquents mais factorisables, forcément peu spécifiques (liaison calendrier, gestion de fichiers rattachés, ...)
- **3 listes du lot 3** : des types très précis, avec haut niveau de service,
- **2 listes du lot 4** : ce lot est libre, vous devrez donc trouver au moins 2 idées de listes ; la richesse des services spécifiques, et l'intérêt fonctionnel des formats proposés entrent évidemment en critère de notation !

LOT 1	LOT 2	LOT 3	LOT 4
Liste de bookmarks sans aperçu	Liste d'anniversaires/événements juste liée au calendrier	Liste de bookmarks avec aperçu	
Liste d'anniversaires ou événements sans système de rappel	Liste d'amis avec anniversaires dans calendrier et rappels	Liste d'anniversaires ou événements avec système de rappel	

## Énoncé de projet de fin d'année

### Outil TOOTLIST

Album photo	Liste de lecture en ligne (morceaux sur Deezer ou autre) avec lecteur embarqué sur le site	Album photo avec zoom, légendes et animations
Liste d'envies/idées/résolutions	Flux RSS	Liste d'amis avec messagerie
Liste de courses	Liste de documents de travail avec droits de lecture/écriture sélectifs	Liste de MP3 avec lecteur custom et stockage
Répertoire téléphonique	todoList, liée au calendrier	Liste de vidéos avec lecteur custom et stockage
Liste d'amis sans messagerie		Liste de programme TV construite depuis site consacré (parsing des infos d'un autre site)
Liste de lecture en ligne (morceaux sur Deezer ou autre)		Simili repository (archivage, versioning, etc...)
Liste de programmes TV saisie en dur		Liste d'évènements datés, avec frise chronologique, et illustrations
Liste de définitions		Liste de tâches avec diagramme de GANTT
Liste de taches simple		

#### Listes dynamiques :

La gestion de listes dynamiques (format ouvert, en plus des formats imposés ci-dessus) est **imposée**. En effet, un utilisateur doit pouvoir générer des listes vides, sans structure particulières, et leur donner catégories, sous-catégories, titres, contenus. Il serait appréciable dans ces listes dynamiques de pouvoir utiliser les éléments de liste déjà élaborés, comme un bloc MP3 ou un lien du calendrier, ...

#### Modèles de listes :

Une fois une liste créée, elle devient un **modèle** de liste : toute liste créée depuis cette première en récupère les propriétés, et vient ajouter son contenu, et rayer /mettre de côté ceux qu'elle ne souhaite plus utiliser. Nous verrons par la suite que ces modifications pourront (suivant un modèle communautaire) être remontées à l'auteur de la liste de départ comme amélioration possible.

Ainsi, tout ce qui se crée étant modèle, la page d'accueil, dans sa partie fixe, proposera une palette de modèles, permettant d'amorcer les listes sélectionnées pour la zone variable.

Il va de soi que la structure (niveaux, titres, nombre d'items etc...) dépend grandement du type de liste et du contenu nécessaire à celle-ci.

Il doit être possible donc de créer une liste depuis un modèle vide (une structure, exemple : Nouvelle Liste de Courses), en dupliquant une liste avec toutes ses données (exemple : liste de courses de Michel > Dupliquer...), et d'autre part, pendant l'utilisation quotidienne de la liste, de la mettre à jour en intégrant les dernières modifications apportées au modèle.

## Énoncé de projet de fin d'année

### Outil TOOTLIST

#### Outils de recherche et catégories:

Lorsqu'une liste est créée, il doit OBLIGATOIREMENT être possible de la relier à des catégories, voire à des sous-catégories liées à des catégories, ... Il doit exister une **hiérarchie de classement**, permettant de trier les listes. Il doit être possible d'attribuer à la liste un résumé ou une liste de **mots-clés** : tous ces éléments d'organisation doivent déboucher une fois les données saisies sur la mise en place d'un outil de recherche.

Vous devrez intégrer un **outil de recherche** puissant vous permettant de rechercher dans les listes d'un utilisateur, dans les listes par défaut ou dans les listes publiques de la communauté (cf juste en dessous).

#### **Authentification / aspect communautaire**

#### Statuts d'utilisateurs :

Comme sur tout bon site avec un front et un back office, vous allez devoir implémenter un système de gestion de droits : il existera 4 types d'utilisateurs :

- **Administrateur** : a accès à toute la plate-forme front-office, mais surtout **backoffice**, seul accès à la configuration de la plateforme ; le sujet n'impose rien de particuliers pour cet utilisateur, mais il devra concentrer toutes les configurations un peu poussées et généralement liées à la plateforme **TootList**.
- **Visateur** : Monsieur Tout-le-monde, internaute lambda qui découvre le site. Il doit pouvoir créer un compte utilisateur et visionner quelques données de base afin de se faire un avis sur la plate-forme. La création des listes lui est évidemment interdite.
- **Membre** : l'utilisateur authentifié, qui a un compte sur **Tootlist**. Il crée ses listes, depuis rien, depuis des modèles, peut faire des recherches, voir les contenus communautaires, remonter ses modifications, configurer contenus et mise en page de sa page d'accueil.
- **Modérateur** : un membre spécial, qui doit pouvoir, via le back-office, valider ou rejeter les propositions d'ajout de contenu public ou de complément sur les existants par les membres. Il doit aussi de manière générale pouvoir faire des modifications sur un compte de membre, et en notifier le propriétaire.

#### Communauté d'utilisateurs:

#### **Vous devrez gérer un système de listes de contacts, inclus d'office dans le pack Membres !**

Chaque liste sera ainsi rattachée à un compte. Le propriétaire d'une liste pourra être ajouté à la liste de contacts des autres (la demande lui sera soumise, à lui d'accepter/rejeter). Le membre peut dès lors choisir si chacune de ses listes est publique (visible par tous les membres), protégée (accessible à ses seuls contacts) ou privée (pour lui-même seulement). De plus, il pourra déterminer si celle-ci est utilisable comme modèle (seulement protégées et publiques).

La communauté a pour but de faire circuler et évoluer les modèles de liste. Un exemple concret :

Un **modérateur** crée une liste des « serial-killers qui marquèrent l'histoire ». Un membre veut récupérer cette liste comme modèle et la compléter : il la **duplique**, ce qui lui offre une **copie**, à laquelle il donnera le nom voulu, et reconnaissant comme **source** la liste originelle. Notre membre modifie sa « list de tueurs kewl » comme il l'a nommée et y ajoute un nom. Une fois cela fait, il se dit qu'il doit à tout prix faire partager son idée : il dispose d'une fonctionnalité « **soumettre comme mise à jour du modèle** ». Une fois cette requête effectuée sur sa liste, il est mis en attente, jusqu'à ce qu'un modérateur lui confirme la prise en compte ou lui notifie le rejet !

## **Énoncé de projet de fin d'année** **Outil TOOTLIST**

La volonté de notre site de s'inscrire dans un mouvement de nature COMMUNAUTAIRE sera favorablement soulignée par le recours à des outils portant déjà cette image : pourquoi ne pas laisser la gestion/représentation des calendriers à un module google ? Utiliser des librairies libres pour les lecteurs MP3 ? Offrir des services depuis le site vers des protocoles célèbres, tels Twitter, Facebook, jabber, ... selon utilité !

### **Sécurité de l'application web**

Un chapitre très bref mais important : il n'est pas envisageable de fournir une application avec une sécurité douteuse ! Vous devrez faire en sorte que vos formulaires rejettent toute forme d'inclusion, que les rafraichissements de pages ne fassent pas de doubles saisies, que la base ne se charge pas de doublons, etc. ... il vous sera demandé de détailler les efforts faits dans ce sens !

### **Application :**

Le « client lourd » ou application embarquée, a pour but de forcer la communication entre votre application web et quelque chose de plus classique.

Nous laisserons volontairement de grandes libertés sur la forme de cette application. Ainsi, il peut s'agir d'un module .NET sur un PC, d'une application iPhone pour ceux qui rêvent de se pencher sur ObjectiveC, du XUL, du WPF, du C++ avec QT, etc. ... Ceux souhaitant bosser sur un support (applications iPhone par exemple) devront venir en parler au plus vite, afin que nous puissions valider ou rejeter les idées proposées, et en cas de confirmation nous procurer le matériel nécessaire pour faire avancer vos développements sur de vrais supports.

Cette application n'a qu'un but concret : être reliée à un compte de membre, charger une de ses listes, voir plusieurs, et exploiter cette information en mode déconnecté ; l'utilisateur doit pouvoir modifier la liste à volonté et remonter les modifications le moment venu (synchronisation). Vous pouvez implémenter cela à la main ou vous appuyer sur des librairies consacrées à la gestion d'informations en mode asynchrone.

### **Contraintes techniques imposées :**

Vous construisez une solution vaste et qui devra se montrer la plus solide possible. Vous serez évalués sur votre créativité, la rigueur de vos choix techniques, mais également sur votre respect des consignes. Un certain nombre de contraintes sont donc imposées :

#### Aspects fonctionnels :

1. Vos membres devront disposer d'un QUOTA : qu'il soit décompté en nombre de listes, en entrées uniques, ou en espace disque, ce quota doit pouvoir permettre de réguler la consommation des ressources de stockage de votre site.
2. Vous devrez à tout prix gérer les listes standards, afin de toucher obligatoirement à une problématique de formulaire dynamique.

## Énoncé de projet de fin d'année

### Outil TOOTLIST

3. Tous vos rendus visuels de listes devront posséder un format imprimable (css de print par exemple ...)
4. Le rendu visuel du site devra être soigné, ergonomique.
5. Vous devrez implémenter une aide utilisateur (quelle qu'en soit la forme !)
6. Un maximum de types de listes doivent pouvoir relier leurs éléments à un calendrier : celui-ci peut s'appuyer sur des produits existants, ce n'est pas un problème. Ex : calendrier des Saints, anniversaires d'une liste de contact, dates sur les points d'un ordre du jour ou les tâches d'une toodolist.
7. Le site doit contenir un certain nombre de données par défaut : un calendrier avec les fêtes, une liste des 10 meilleures listes, une liste de recettes, etc. ... qui puisse servir de modèles de base lorsque le site n'est pas ou peu rempli.

#### Aspects architecturaux :

1. La communication entre l'application autonome et le site se fera à travers une structure de WEB SERVICES, libre à vous d'en déterminer la forme, que vous devrez présenter.
2. L'application, hors site web, ne doit pas s'appuyer sur un serveur !

#### Aspects de sécurité:

1. Pensez à soigner la qualité de votre application web : on ne peut pas saisir n'importe quoi et tout casser, le rendu doit donc être souple et la saisie protégée contre l'inclusion.
2. Toute donnée destinée à l'affichage public doit être soumise à modération !
3. L'accès au back-office devra se montrer le plus sécurisé possible : préparez vous à justifier vos choix.
4. Afin d'éviter les erreurs de parcours, vous veillerez à distinguer visuellement les zones en front et en back-office.

## Fonctionnalités bonus

Ces fonctionnalités « bonus » ne le sont pas vraiment, puisqu'il s'agira d'en implémenter un maximum. Plus votre effort pour en implémenter (correctement) sera grand, meilleure sera la note. Ces projet est une application, pas un concours de code ! Tachez d'être orientés FONCTIONNEL et SERVICES. Ainsi, piochez à volonté dans la liste suivante ou proposez vos propres extensions sur le sujet de départ :

- Liste d'adresses avec localisation sur une carte de l'adresse.
- Internationalisation (rendre le site disponible dans plusieurs langues)
- Chat, messagerie instantanée, mail...
- Une partie de plate-forme consacrée à l'utilisation en milieu professionnel, centrée sur les notions de travail communautaire, partage de documents, avec commentaires associés, verrou de données etc...
- import/export au format XML de mes données de listes
- Adresse mail attribuée à chaque liste pour pouvoir y ajouter des éléments ou simplement la consulter (du push d'informations via SMTP), voire un moyen de gérer ses listes via un client mail !
- Gestion à partir du back office des insertions publicitaires
- Gérer différentes priorités pour chaque tâche ou listes
- Possibilité d'éditer et donc de sauvegarder les modifications sans recharger la page (couche Ajax)
- Rendre le site disponible dans plusieurs langues
- Mettre en place un système de notification de tâches par mail, SMS ou encore messagerie instantanée et cela sur presque toutes les plateformes.
- Possibilité d'intégrer les listes sous Netvibes, Google Agenda ou Facebook.
- Partage d'information sensible (stockage distants : ex info perso, codes, mot de passe...) : mettre en place une politique de sécurité spéciale (ssl, encryptage, systèmes de confirmation, saisie de mots de passe avec captcha.
- Proposer un rendu de liste très spécifique (rendu pour écran de smartphone, l'écran d'un iPhone, un format WAP, ...)

# ***PROJET ING3 SIGL***

Énoncé de projet de fin d'année

*Outil TOOTLIST*

## ***Modalités de rendu et notation :***

Tout cela sera détaillé le moment venu.