

PROJET WEB APP

CAHIER DES CHARGES

Médéric NETTO – Kévin PEETERS

TABLE DES MATIERES

Table des matières

I – Analyse du besoin	1
II – Fonctionnalités proposées	3
III – Choix techniques	4
IV – Product backlog	5
V – Équation délais / budget	6
VI – Bilan	7

I – Analyse du besoin

L'entreprise Takatoukité spécialisée dans le développement web a récemment mis en place un nouvel ERP au sein de ses équipes.

Les demandes d'interventions (DI) sont fréquentes et justifient amplement l'utilisation d'un outil de ticketing. Cet outil sera développé en interne par l'équipe de développement de l'entreprise composée d'un chef de projet et de deux développeurs.

Les membres de cette équipe ont ainsi été missionnés par la Direction pour réaliser un prototype : ils disposent de trois jours pour développer un POC (Proof Of Concept) qu'ils devront présenter aux termes de ce délai.

RAPPORT PROJET WEB APP

1 – LEAN CANVAS

Problème	À la suite de la mise en place du nouvel ERP le service informatique de la société Takatoukité fait face à une demande constante du nombre d'interventions
<i>Alternatives existantes</i>	GLPI, JIRA, Mails
Segments de clientes	Administrateurs systèmes
<i>Utilisateurs pionniers</i>	Membres du jury, échantillon d'admin. systèmes
Proposition de valeur unique	La solution est réalisée sur mesure et n'embarque pas les éventuelles autres fonctionnalités inutiles proposées par la concurrence
<i>Votre "Pitch" !</i>	Takatou-ticket
Solution	<ul style="list-style-type: none">- Afficher la liste des demandes d'interventions sous la forme d'un tableau- Filtrer la liste des interventions par colonne- Ordonnancer les interventions par colonne- Afficher les détails d'une intervention donnée- Ajouter une intervention- Éditer en détails une intervention- Éditer rapidement les informations principales d'une intervention- Supprimer une intervention- Paginer les résultats- Rechercher de manière globale n'importe quel mot clé relatif à une intervention
Avantage compétitif	Outil développé en interne donc au plus proche des besoins du client
Sources de revenus	Projet financé en interne
Coûts	Salaires + Frais d'hébergement de la solution
Indicateurs de performance	Baisse du nombre d'interventions actives
Canaux	POC

II – Fonctionnalités proposées

La Direction désire que les fonctionnalités listées ci-dessous soient implémentées dans la solution définitive :

- Lister les DI :
 - Sous forme d'un tableau
 - Paginer les résultats : 50 DI par page
- Filtrer la liste des DI :
 - Les filtres seront appliqués aux colonnes du tableau
 - Un ou plusieurs filtres pourront être appliqués simultanément
- Trier les DI :
 - Les tris seront appliqués aux colonnes du tableau
 - Une ou plusieurs colonnes pourront être triées simultanément
 - L'utilisateur décide dans quel ordre sont triées les colonnes
- Visualiser une DI en détails
- Ajouter une DI
 - Renseigner obligatoirement les informations suivantes :
 - L'utilisateur à l'origine de la demande
 - La catégorie
 - Un titre
 - Une description
 - Le degré d'urgence : très faible, faible, normale, élevée, très élevée
 - Renseigner éventuellement :
 - Le technicien en charge de la DI
 - Valeurs par défaut :
 - Statut : nouveau
 - Date d'ouverture : date à l'enregistrement
 - Urgence : normale
- Modifier une DI
 - La date de clôture sera automatiquement renseignée lorsqu'une DI sera statuée comme étant « terminée »
- Effectuer une modification rapide sur une DI
 - Les champs suivants doivent pouvoir être modifiés rapidement :
 - Statut
 - Le technicien en charge de la demande
- Supprimer une DI
- Rechercher une DI :
 - Par mots clés

III – Choix techniques

1 – HTML / CSS

Nous souhaitons mettre en place une solution rapide à développer et mettre en place, et peu coûteuse ; c'est pourquoi notre choix s'est naturellement porté vers une solution web. Nous construirons ainsi à l'aide du langage de balisage HTML et de feuilles de style CSS un site responsive multiplateformes.

2 – JAVASCRIPT

Nous nous servirons de Javascript pour dynamiser notre site et le rendre davantage interactif.

3 – FRAMEWORKS

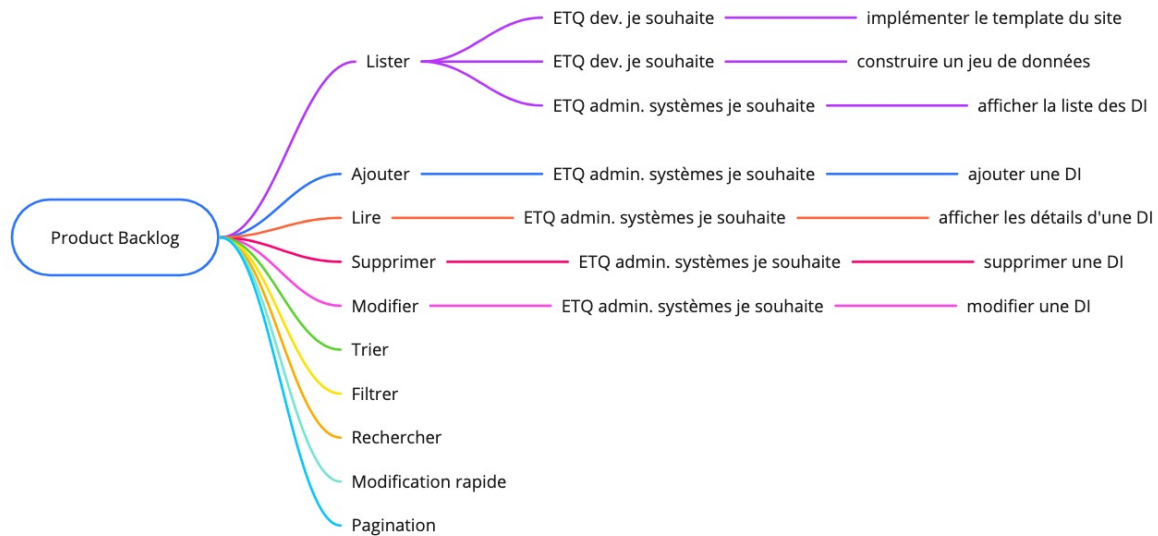
Nous avons opté l'utilisation de deux frameworks :

- Le framework Javascript React :
 - Très populaire :
 - En 2018, sur LinkedIn, 70k emplois orientés autour de React
 - Communauté grandissante :
 - Sur le projet Github, 140k étoiles en 2020 vs 120k en 2019
 - Très bien documenté
 - L'équipe est habituée à travailler avec des frameworks Javascript : Angular, React, Vue
- Le framework CSS Material-UI :
 - Permettra la réutilisation de composants simples déjà existants de manière à ne pas gâcher inutilement du temps à les réimplémenter
 - L'équipe est habituée à travailler avec d'autres frameworks similaires, en particulier Bootstrap

4 – GESTION DOCUMENTAIRE

- Le POC sera versionné à l'aide de l'outil de versioning Git ; les sources seront stockées sur un repository Github.
- Les documents sont stockés sur un SharePoint :
 - Permet aux membres de l'équipe de travailler simultanément sur les documents texte et feuilles de calculs
 - Ces documents seront régulièrement copiés sur le repository Git à la demande du chef de projet

IV – Product backlog



miro

Notre Product backlog se présente sous la forme d'une carte mentale réalisé avec l'outil en ligne miro.

Les différentes fonctionnalités sont classées par ordre de priorité, nous sommes partis sur le fait que pour le POC il serait essentiel à l'utilisateur de pouvoir réaliser un CRUD soit ;

- Créer
- Lire / Lister
- Modifier
- Supprimer

Que l'on a ensuite classé par ordre de priorité. Partant du principe qu'il y ait déjà des données on a préféré commencer par lister les données plutôt que d'en créer qui sera prochaine puis lire, supprimer et modifier qui permettrons vraiment à l'utilisateur de jouer avec le produit.

Contrairement aux fonctionnalités : trier, filtrer, rechercher, etc. qui sont des fonctionnalités non essentielles que l'on pourrait voir apparaitre dans de future version.

V – Équation délais / budget

Le budget du POC sera alloué par l'entreprise

1 – DEVELOPPEURS

Les couts du POC en fonction du délai est estimé à **675€** brut pour deux développeurs en sachant que l'on travail 7h par jour.

Avec un salaire de 30k brut / an soit 2.5k brut / mois

Avec temps de travail de 151h / mois

Qui donne : $2500 / 151 = 16\text{€}/\text{h}$

$16 \times 7 = 112\text{€}$ pour une journée de 7h

$112 \times 3 = 336\text{€}$ pour 3 jours de développement pour un développeur

Soit $336 \times 2 = 672\text{€}$ brut pour 2 développeurs

2 – HEBERGEMENT

Les couts liés à l'hébergement peuvent être nul si l'entreprise possède déjà ses propres serveurs.

En revanche le cout d'un hébergement en ligne sur des plateformes tel que OVH, PlanetHoster et Infomaniak sont d'environ **7€** TTC par mois.

VI – Bilan