Cahier des charges fonctionnel pour le Travail Pratique 3- Manolo Racine

1. Introduction au projet

**Entreprise**

- *Nous sommes une bibliothèque qui a pour but de simplifier et d’automatiser l’emprunt de différents documents grâce à un système informatique façile d’utilisation*

**Typologie de produit**

- Nous voulons créer une application web qui fonctionne sur la majorité des navigateurs internet qui pourrait être utilisé en tant qu’application mobile pour Android et iOS

**Objectifs du produit**

Nous voulons lancer ce projet à cause du grand nombre de pertes, difficultés et complications liées avec l’utilisation d’un système manuel

L’objectif de ce produit est de réduire les pertes d’argent dût aux amendes perdues, rendre les opérations plus faciles pour les clients et les employés et réduire les pertes de livres

Nous sommes inspirés par les sites webs de différentes bibliothèques publiques, car ce sont déjà plateformes établies et qui ont fait leurs preuves

Le produit est un nouveau produit fait spécialement pour cette bibliothèque

Le produit est une plateforme qui permet à des clients de d’emprunter des livres et de payer leurs amendes, qui permet aussi aux préposées de d’ajouter des documents dans le système et aux gestionnaires de voir différentes statistiques sur les opérations de la bibliothèque

**La valeur ajoutée du produit**

La promesse du produit est de réduire ou éliminer le temps qui était auparavant utilisé pour s’occuper des problèmes de gestion qui viennent et d’offrir une interface d’emprunt plus conviviale au client

**Le modèle économique**

Le produit va être financé par la ville où la bibliothèque se trouve et la valeur ajoutée sera mesurée par le nombre d’emprunts et le nombre d’amendes ou de livres perdus. Une source de revenus secondaire proviendra des amendes de retard collectés.

**Pour qui ?**

Les utilisateurs du produit seront :

Un client/emprunteur : un client de la bibliothèque qui veut emprunter de livres et qui devra les retourner avant une certaine date. Il doit payer des frais de retard lorsqu’il retourne des livres en retard. Il doit pouvoir accéder à page/application qui lui indique les différentes informations sur ses emprunts et sur son compte. Il veut aussi chercher des documents dans l’application basé sur différents critères.

Un préposé : un employé de la bibliothèque qui doit ajouter différents types de documents ainsi que leurs informations à la base de donnée de l’application.

Un gestionnaire : une personne qui veut interroger le système pour connaitre certaines informations sur les statistiques de la banque comme le nombre d’emprunts et le nombre d’amende.

**Perspectives de développement**

Le succès du produit sera mesuré par la quantité de personnes qui utilise la bibliothèque augmente et que le nombre d’amendes ou de documents perdus est réduit à près de 0

Le nombre d’utilisateurs actifs serait environ 3% de la population de la ville dans la laquelle se trouve la bibliothèque

Les prochaines étapes du projet seraient de corrigé les bugs qui seront trouvés lors de la mise en production et d’ajouter des fonctionnalités à la demande du client

2. Tableau fonctionnel

Ce tableau résume vos besoins fonctionnels sous forme de ‘user story’. Il ne s’agit pas de remplir des solutions techniques, mais bien des besoins. Voici les champs que vous aurez à remplir :

* **Story : votre besoin ;**
* **Critères d’appréciation : les critères par lesquels la bonne réalisation du besoin sera mesurée ;**
* **Niveau de performance : le niveau visé pour chaque critère listé.**

Priorités

Quelles sont les fonctionnalités prioritaires ?

Nous vous invitons à hiérarchiser par ordre d’importance et d’urgence les fonctions que vous voulez que votre produit remplisse.

Un ‘story map’ peut être utilisé pour mieux définir les priorités. (Tel que vu au cours 420-535)

**Comment prioriser avec la méthode MoSCoW ?**

Ce Framework qui fait partie des méthodologies agiles permet de vous faciliter le travail de priorisation et de hiérarchisation de vos besoins.

* **Must : fonctionnalités essentielles et non négociables.**
* **Should : fonctionnalités à implémenter dans la mesure du possible.**
* **Nice-to-have: fonctionnalités qui contribuent à la satisfaction de l’utilisateur mais qui ne sont pas essentielles.**

Liste des ‘User stories’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Priorité | En tant que… | Je veux… | Afin de… |
|  |  | Client | Chercher un article   (document, livre, cd ou dvd). |  |
|  |  | Client | Obtenir la disponibilité d’un article. |  |
|  |  | Client | Emprunter un article. |  |
|  |  | Client | Retourner un article |  |
|  |  | Client | Voir la quantité disponible pour un article. |  |
|  |  | Client | Voir la condition d’un article. |  |
|  |  | Client | Voir tous les articles empruntés en ce moment. |  |
|  |  | Client | Voir les dates limite des emprunts. |  |
|  |  | Client | Voir les frais de retard. |  |
|  |  | Client | Voir l’historique d’emprunt. |  |
|  |  | Client | Obtenir des recommandations en fonctions de l’historique d’emprunt. |  |
|  |  | Client | Payer les frais de retard |  |
|  |  | Préposé | Ajouter un article à la banque d’articles. |  |
|  |  | Préposé | Modifier un article. |  |
|  |  | Préposé | Voir les articles présentement dans la banque d’article et leur quantité. |  |
|  |  | Préposé | Supprimer un article. |  |
|  |  | Gestionnaire | Voir les articles présentement dans la banque d’article et leur quantité. |  |
|  |  | Gestionnaire | Interroger le système selon un besoin spécifique. (Exemple : savoir le nombre de documents emprunter par mois.) |  |
|  |  | Gestionnaire | Voir l’historique d’emprunt d’un client. |  |
|  |  | Gestionnaire | Voir tous les emprunts courants, leur date, et les clients. |  |
|  |  | Gestionnaire | Voir l’historique d’emprunt d’un article. |  |

3. Contraintes graphiques

**Valeurs et sources d’inspiration**

Nos inspirations pour le design graphique sont les sites de bibliothèques municipales et nous voulons mettre de l’avant la fonctionnalité d’emprunt de documents.

4. Contraintes techniques

**Architecture**

Nous n’avons pas de contraintes d’architecture ou d’API déjà utilisables

Les données du produit seront hébergées dans une base de données h2. L’application et la base de données seront hébergées sur un serveur à l’intérieur de la bibliothèque

L’application sera améliorée ou changée à la demande du client.

**Spécificités Techniques**

L’application utilisera Java comme langage de programmation, les données seront traitées dans une base de données H2 et l’application sera maintenu lorsque des bugs ou des problèmes seront découverts.

L’application sera une application web, donc elle devrait fonctionner sur la majorité des appareils ayant accès à un navigateur web standard

Tous les navigateurs seront supportés, sauf les navigateurs qui n’ont pas été mis à jour depuis longtemps ou qui sont très peu populaires (ex : internet explorer), ce qui permettre à l’application d’être utilisé sur des ordinateurs, téléphones intelligents et la plupart des appareils ayant accès à l’internet.

5. Encadrement du projet

**Typologie de la gestion de projet**

Le type de gestion de projet utilisé sera le Scrum.

Le design sera conçu à partir de zéro, sans modèle préexistant autre que des inspirations prises d’autres sites

L’équipe travaillera sur place (Puisqu’il y a seulement une personne dans l’équipe)

Il y a seulement une personne qui travaille sur le projet, donc toutes les responsabilités seront sur cette personne

ANNEXE. Diagrammes UML et schémas de base de données.

