Cahier des Charges fonctionnelles – Application SaaS de gestion d'association

Wassim Bech IA2.1.1 ENISo 2013

1- Présentation du sujet

1.1 - Problématique

Il s'agit de développer une application qui aidera à la gestion et l'administration de l'association estudiantine ENISo Junior Entreprise. Elle devra automatiser certaines tâches et sera utilisée par plus d'une cinquantaine d'utilisateurs.

Actuellement la dite gestion s'effectue via des google docs et vu le nombre important de contributeurs de l'association, cela induit des problèmes de synchronisation et de transmission de l'information.

1.2 - Les objectifs

L'application doit permettre la synchronisation immédiate des différentes tâches effectuées dans l'association.

Elle doit communiquer avec le site internet et les bases de données hébergées de l'association.

2- Spécifications des besoins

2.1 Besoins fonctionnels

- Gestion des tâches
  S'inspirant des logiciels de gestion de projets, l'application permettra la création d'une tâche avec tous les détails relatifs ; description de la tâche, date butoir, ressources engagées. Egalement, la tâche devra avoir un responsable (généralement son créateur) et ce dernier peut assigner d'autres utilisateurs. Le responsable et les utilisateurs sont les membres de l'association et sont énumérés dans une base de données.

- Master-list
  Les tâches en cours doivent être synchronisées et regroupées dans une seule liste. Cette dernière devra être envoyable au site web et téléchargeable de manière rapide et transparente par le programme.

- Système de Mérite
  L'association classe ses membres selon un système de mérite favorisant les plus contributeurs. Il octroi des points à chaque tâche réalisée en stipulant un résumé de la tâche, sa date et un justificatif. Le programme devra automatiser cette tâche (octroi de point sur base d'accomplissement d'une tâche). Pour ce faire il devra se référer à une grille présentant le nombre de points donnés par tâche. Cette grille peut être modifiée par le bureau exécutif de l'association.

- Auto-gestion des membres
  Les membres auront accès à la master-list et décideront quelle tâche rejoindre ou créeront leur propre tâche. Des notifications devront donc faire un compte rendu de ces rotations de l'effectif aux membres concernés. Egalement, un système de compte-rendu sur l'avancement des tâches est prévu.

2.1- Besoins non fonctionnels

- L'application devra être intuitive, élégante et ergonomique pour faciliter au mieux l'autogestion.

- La sécurité des transmissions des données entre l'application et les bases de données devra être sans failles.

2.3- Utilisateurs

Il y aura deux niveaux d'utilisateurs, qui reflètent la structure de l'association :

A –Accès responsable (également administrateur) : restreint au bureau exécutif de l'association. Ils pourront modifier ou supprimer l'ensemble des tâches, même celles qui ne leur appartiennent pas. Ils ont aussi un droit d'administration sur les utilisateurs de l'application. (Ajout, suppression, promotion au grade de responsable)

B – Accès membre : Ils pourront proposer, modifier et supprimer à volonté les tâches qui leur appartiennent. Ils peuvent à tout moment consulter et interagir avec celles des autres mais sans les modifier.

3. Environnement de travail

3.1- Environnement Matériel

WEBrick, serveur libre intégré à Ruby pour le développement.

Apache, nginx ou Mongrel sont conseillés pour la production.

3.2 – Environnement logiciel

Framework de développement suggéré : Ruby on Rails, Ruby étant un langage interprété et également orienté objet.

IDE suggéré : RubyMine.

4- Chronogramme

<table>
<tr><td colspan="9"></td><td>Déploiement et tests</td></tr>
<tr><td colspan="4" rowspan="2">Etude de l'existant et apprentissage de Ruby</td><td colspan="5">Réalisation de l'application</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Fev 1</td><td>Fev 2</td><td>Fev 3</td><td>Fev 4</td><td>Mars 1</td><td>Mars 2</td><td>Mars 3</td><td>Mars 4</td><td>Avr 1</td><td>Avr 1</td></tr>
</table>