Cahier des charges : Hop3x Full-Web

Groupe 5 M1-ISI

13 janvier 2016

1 Presentation d'ensemble

- Un Site Web Environnement de Developemment destiné a l'enseignement.
- Le site devra permettre la sauvegarde automatique des fichiers.
- L'utilisateur pourra compiler et executer ses programmes, et voir le flux sortant dans un terminal.
- Plusieurs classes d'éleves pourront travailler en meme temps sur leurs fichiers.
- Les professeurs peuvent visualiser des informations sur leurs eleves.
- Les professeurs peuvent écrire des tests qui seront jouées par les étudiants sur leurs projets universitaires.
- Les éleves devront pouvoir créer leurs propres.

Le but du projet est de creer la refonte full web de ce logiciel qui existe déja en logiciel lourd. La raison de ce projet est d'éviter a l'utilisateur a installer le moindre composant logiciel (comme java) sur son ordinateur.

2 Charte graphique :

Inserer les mockups finalisé
Il manque des informations (Couleurs/Images/Arrondis/Ombres/Interactions/Importance de la responsivité/etc).

3 Specifications fonctionnelles

3.1 Editeur de texte

Coloration syntaxique :
L'éditeur devra supporter la coloration syntaxique des fichiers pendant leurs dévellopement.
Fonction de recherche/remplacement :
L'utilisateur pourra entrer une chaine de caractère dans un champ, en entrer une deuxieme dans un autre champ, et l'application devra effectuer une permutation de ces 2 chaines dans le code situé dans l'editeur.

3.2 Gestion persistances de données

Traces :
Le site devra stocker chaque évenement sur les fichiers et les trier.
Reconstruction des fichiers :
Le site devra trouver et reconstruire un fichiers d' utilisateur en fonction de qui il est et de quel project il fait parti.

3.3 Projets de differents types

plusieurs languages disponibles :
L'utilisateur pourra créer des projets dans différents languages, nottamment C/Ruby/Java/Python
L'utilisateur doit pouvoir compiler/executer sans avoir a entrer une ligne de commande,mais il devra pouvoir ajouter des arguments (au main)

3.4 Compilation et execution sur le serveur

Output redirigé dans le terminal du client :
L'utilisateur a accès en lecture a un terminal qui lui indique les messages de retour de ses logiciels, ou du compilateur.
**pas d'ui possible a ma connaissance :
Vérifier si c'est grave, mais je pense que cela sort de l'utilité principale,
au pire nous pourrions inclure un lien vers l'ancien hop3x,
en precisant que la fonction User Interface est indisponible sur le site.

3.5 Gestions de 3 types d'utilisateurs (Profs/Eleves/Administrateur)

Administrateur :
L'administrateur pourra créer/modifier/Supprimer des professeurs commes des éleves.
Professeurs :
les profeseurs pourront créer des nouveaux comptes d'éleves, les modifier, et les supprimer.
Les professeurs pourront créer des classes, ajouter des éleves dedans, et assigner des projets avec tests pour cette classe.
usernames generées en fonction des infos :
Lors d'un ajout d'un nouvel utilisateur, le site doit généré un username en se basant sur les infos récupéré, en évitant les doublons.

3.6 Informations sur les utilisateurs :

***La liste est encore incomplete et incertaine
Le professeur aura acces a une interface pour visualiser des informations automatiquement générés sur ces éleves.
Il verra si l'utilisateur est actuelment actif, connecté(mais inactif, la différence pourrait se faire par rapport a un temps sans evenement)ou deconnecté
Il pourra visualiser les Projets/Repertoires/Fichiers de ses éleves, ainsi que si besoin d'autre infos (date de créa, date de modification, nombre de compilation).
Il aura acces a un mode de lecture de la facon dont l'éleve a écrit son code :
en temps reel(vitesse d'écriture réele de l'étudiant)
ou en live (quand l'éleve est actuelement en train de travailler sur son fichier)
ou evenement par evenement(d'apres les cliques sur le boutons adéquat)
(essayer de gérer le cas ou l'utilisateur change de fichier)

3.7 Vues sur les tests et les resultats des eleves :

Le site permettra a l'utilisateur de lancer une compilation/execution différente, car incluant un test selectionné par appui sur un bouton.
Les résultats de ces tests devronts être stockés, et ***les informations importantes seront calculés et affiché sur la page de l'enseignat.

3.8 Creation de projets

Creation de tests par les professeurs si projet universitaire :
Les enseignants sont les seuls a pouvoir créer des projets universitaires, et donc a écrire les tests.
Ils pourront assignés ce projet a des classes d'éleves directement.
Tests en code reel ecrit dans le language cible :
Les tests s'ecriront en lignes dans le meme languages que les types de projets a tester.
**Ils s'ajouteront dans le main, ou bien le remplacera,(une annotation laissé par l'etudiant comme "£test£",
permettrai a l'étudiant de le placer lui meme dans son code.)
Projets decoupables en fichiers et repertoire :
Les projets contiendront des fichiers, mais aussi des repertoires, et ceux ci seront créables par l'utilisateur.
**S'il peut les emboiter, il faut réflechir a une UI qui permettrai un affichage efficace, et peu couteux.

4 Specifications techniques

- Le serveur sera lancé sur un pc qui possède un systeme UNIX (actuelement Mac-Os (?))
- Acces concurents sur les fichiers (ex : un eleve et un prof en mode live), nous avons opté pour le SGDB MySql qui gère nativement les accès concurents.
- Le choix des technologies doit être assez simple mais efficace car d'autres éleves pourront être amenés a travailler sur un upgrade du site l'an prochain.
- La documentation doit être complête (pour les futurs dévellopeurs), malgré que le projet suive la méthode Scrum , qui veut que la production prime sur la documentation.
- Le serveur devra supporter ***N utilisateurs actifs en meme temps.