Sujet sur le pdf :

Le logiciel doit pouvoir superviser le recrutement des étudiants, qui peuvent être recrutés sur **plusieurs concours** : un concours destiné aux élèves en classes préparatoires, et un autre aux élèves titulaires d’un diplôme ou d’une équivalence (dits “sur titres”).

Le concours CPGE est géré par le SCEI qui transmet les informations nécessaires à l’école une fois les candidats sélectionnés.

Le concours sur titres est organisé par l’école et piloté à l’aide du présent logiciel. Il doit notamment permettre aux candidats de déposer les pièces constituant leur dossier, à l’école de déclarer admissible certains candidats, d’organiser des entretiens pour les évaluer puis de déclarer admis certains candidats. Le suivi et l’intégration des candidats doivent également être possibles. L’objet du recrutement est de permettre ensuite l’accueil des étudiants pour l’intégration de l’école à la rentrée.

Users :

* candidats ;
* diplômés ;
* direction des études ;
* enseignants ;
* étudiants ;
* gestionnaires concours ;
* gestionnaires relations entreprises ;
* gestionnaires ressources humaines ;
* gestionnaires scolarité ;
* responsables d’UE ;
* responsables de diplôme ;
* responsables de module ;
* responsables pédagogiques.

**Questions à poser lors de la première réunion avec la MOA :**

FORME :

* interface web avec le candidat et gestionnaire de concours logiciel ?
* ou que un logiciel

*Commentaire (Thibault) : je pense qu’ils vont nous demander de choisir nous même. Le premier point me semble bien adapté MAIS si ce que vous entendez par logiciel, c’est le “backend” autant aussi faire une webview pour les autres utilisateurs du système d’information. Le backend fait les traitements, les recherches dans la BDD, ect… . Comme ça on peut faire une webview accessible de partout pour tous le monde. Donc ça serait une 3ème solution possible*

FONCTIONNALITÉS :

* Déposer les fichiers d’inscription :
  + PDF ? // ça dépend si il faut faire du “parsing” ?
  + quelles sont les pièces justificatives
* Suivre le dossier de candidature :
  + mail …
  + interface ? // ou interface web + mail ?
* Déclarer l’admissibilité
  + interface // ou interface web + mail ?
* Organiser les entretiens
  + info ? date, personnes etc..
  + gestion des entretiens ? cad comment se passe l’organisation des entretien : c’est le logiciel qui va générer des passages ?, choix de plage horaire pour le jury ?
  + envoie des informations de passage par mail, via le logiciel, via le web ?
* Intégrer les candidats
  + en quoi ça consiste ? redirection vers l’accueil // envoie d’un mail / notification d’admission ?
  + (groupe 2) interface scolarité : telecharger des fichiers pour l’inscription ?
* Suivre les candidats
  + quels info pour qui ?
  + en quoi ça consiste ?

USERS :

* Candidats SCEI
* Jury
* Gestionnaire de concours = Admin ?
* Scolarité
* qui à accès à quoi ?

CANDIDATS :

* types de données (prénom, nom, DdN, mail, notes ? )
* Pièces du dossier se limitent à des PDF ou autre ?

SCEI :

* A-t-on besoin des données SCEI à un moment ?
  + (Comment récupère-t-on les données de SCEI ?) (à demander plus tard) // web service ??
  + Quel est le format des données de SCEI ?

**Framework pour le côté GUI :**

JavaFX ( ? ) : Semble être bien plus joli en rendu que Swing. Je ne sais pas si c’est simple d’utilisation mais je suppose que ça ne doit pas être ignoble.

Bootstrap pour un site avec du node.js ?

Backend : mySQL / serveur / nodeJS ou JSP (java) ou php

Front : HTML / CSS / javascript

Approche TDD ??

Fonctionnalités supplémentaires :

* L’application doit permettre d’obtenir **l’accès à des bilans et des synthèses** portant sur les données manipulées et enregistrées propres à chaque sous-projet.
* Afin de garantir le bon fonctionnement de chaque sous-projet, il sera attendu qu’ils puissent chacun être **testés en conditions réelles**, afin de notamment garantir la **compatibilité** avec d’autres outils déjà en place, ou encore les autres sous-projets.
* Les données manipulées par l’application doivent pouvoir être **exportées** vers un format de **type tableur**. De même, l’application doit pouvoir prendre en **entrée** des tableurs dans le cadre de certaines fonctionnalités afin de valider leur fonctionnement.

Livrables :

* Documentation présentant la planification du projet et l’organisation de son équipe.
* Gestion du suivi client
* Liste et organisation des ressources, matérielles humaines et logicielles
* Dossier d’architecture
* Document de conception
* Documentation utilisateur
* Plan de déploiement
* Inventaire des exigences
* Jeux de test
* Résultats de tests
* Protocole de test
* Réalisation de sessions de tests
* Compte rendu détaillé des sessions
* Document de synthèse
* Logiciel fonctionnel