

RAPPORT DE PROJET LONG
INFORMATIQUE

Questionnaire vidéo

Equipe

Huiting LI
Antoine ROUSSEAU

Encadrants

Cécile Braunstein
Ludovic Saint-Bauzel

En partenariat avec l'entreprise

Beetween représentée par Mr Pierre Muset

Année 2010-2011

Préface

Ce document a pour but de décrire le déroulement de notre projet long d'informatique ELI4. C'est le résultat du travail qui nous a permis de réaliser un site internet d'entretien par vidéos interposées.

Ce rapport contient l'ensemble des éléments du projet. D'un point de vue technique tout d'abord, nous présenterons le cahier des charges tel que nous l'avons imaginé, puis les spécifications plus détaillées qui en découlent. Nous décrirons le fonctionnement de notre projet dans son ensemble ainsi que les éléments qui prouvent le bon fonctionnement de celui-ci. Pour terminer la partie technique, nous présenterons nos impressions sur le projet concernant les difficultés techniques rencontrées et les perspectives ouvertes.

La deuxième partie de ce rapport a pour objectif de présenter la manière dont nous avons géré le projet. Nous présenterons dans un premier temps comment le projet a été découpé en tâches afin de mieux se répartir le travail, puis comment s'est organisé celui-ci au sein de l'équipe. Nous présenterons ensuite le diagramme de Gantt tel qu'il a été établi au début du projet, puis ferons le point sur les difficultés rencontrées, avant de présenter le diagramme de Gantt qui a été effectivement réalisé.

La troisième partie a pour objectif de présenter l'ensemble des apports du projet : d'un point de vue technique, gestion de projet, pédagogique, puis personnel.

Nous espérons que vous prendrez autant de plaisir à lire ce rapport que nous en avons pris durant tout le déroulement de ce projet.

L'équipe Questionnaire vidéo

Sommaire

I. Rapport technique

1. Cahier des charges	6
2. Spécifications fonctionnelles et détaillées	11
3. Tests et validation	19
4. Recette	20
5. Performances	21
6. Difficultés techniques et solutions apportées	21
7. Conclusion et Perspectives	23

II. Rapport de gestion du projet

1. Découpage en tâches	24
2. Organisation de l'équipe	25
3. Diagramme de Gantt initial	26
4. Difficultés rencontrées dans la gestion du projet et solutions apportées	27
5. Diagramme de Gantt réalisé	27
6. Conclusion et perspectives	28

III – Apports du projet

1. Apports scientifiques et techniques	28
2. Apport sur la gestion de projet	28
3. Apport sur la formation pédagogique	29
4. Apport personnel	29

Conclusion	29
------------	----

V – Annexes

Captures d'écran du site réalisé	30
----------------------------------	----

Contexte

Ce projet nous a été proposé par l'entreprise Beetween.



Beetween est une société de recrutement. Les recruteurs de la société reçoivent les demandes des entreprises qui cherchent un profil de candidat pour un poste particulier. Beetween se charge de recruter la bonne personne et de la diriger vers l'entreprise demandeuse.

En ce sens, les employés de Beetween reçoivent très souvent des demandeurs d'emploi. Les recruteurs se chargent de mener à bien un entretien pour déterminer si le profil de la personne correspond au besoin formulé par l'entreprise. L'entreprise demandeuse gagne donc du temps en ne s'entretenant qu'avec des candidats précédemment sélectionnés.

Toutefois, les coûts générés par le fait de recevoir des candidats quotidiennement est important, tant en terme de temps qu'en terme financier. L'idée de notre projet est alors venue.

Nous nous proposons de mettre en place un système de questionnaire vidéo, à mi-chemin entre un questionnaire papier et un entretien en face à face. Les questions et réponses se font par vidéo et la personne répondant au questionnaire doit répondre « du tac au tac » à chaque question posée, comme elle le ferait lors d'un entretien en face à face.

Les avantages sont nombreux, tant pour le recruteur que pour le candidat. Le candidat n'est plus obligé de se déplacer dans les locaux de l'entreprise et les recruteurs de Beetween peuvent s'atteler à d'autres tâches. De plus, les problèmes de disponibilité disparaissent. Les deux personnes peuvent consulter l'entretien ou y répondre quand ils le désirent puisque les entretiens ne sont font pas en direct mais par le biais de vidéos enregistrées.

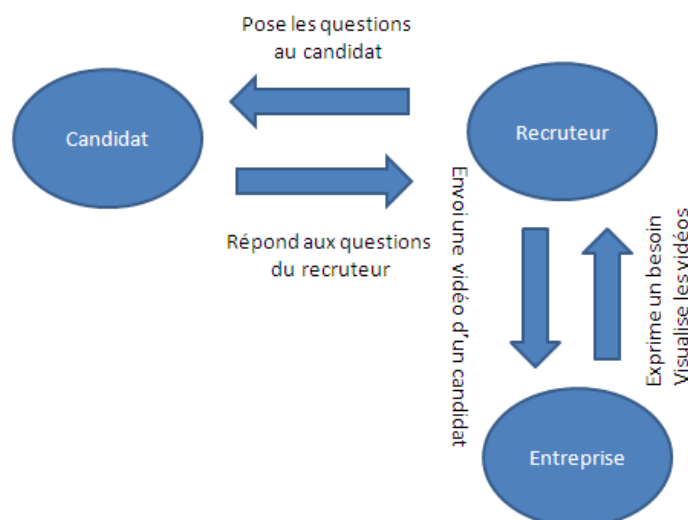
Scénario

Pour bien comprendre le projet, un scénario simplifié est décrit ci-dessous, dans lequel le recruteur A doit envoyer un questionnaire vidéo à un candidat B.

- Le recruteur A se filme. Il enregistre une question puis stoppe l'enregistrement et sélectionne une durée maximale de réponse pour cette question.
- Il réitère l'opération autant de fois que souhaité. Il peut donc enregistrer plusieurs questions à la suite.
- Il ordonne les questions pour former un entretien cohérent.
- Il envoie via email un lien vers la séquence de questions à un candidat B.
- Le candidat B suit le lien reçu par email, et arrive sur la page de l'entretien.
- Il visualise la première question du recruteur A. Quand le recruteur A a fini de parler, c'est au tour du candidat B d'être filmé, et sa réponse à la question du recruteur A est enregistrée. L'enregistrement s'arrête lorsque le candidat B le décide, ou lorsque le délai choisit par le Recruteur A pour cette question est expiré.
- Ce scénario se répète pour chaque question enregistrée par le recruteur A.
- Lorsque le candidat B a répondu à toutes les questions du recruteur A, la session s'arrête. Le recruteur A reçoit par email un lien lui permettant de visualiser la séquence complète, Comprenant les questions du recruteur A et les réponses du candidat B.

On comprend donc qu'il y a 3 acteurs dans ce projet. C'est autour des deux acteurs principaux que sont le candidat et le recruteur que se situe le cœur de l'application. L'entreprise ne fait qu'exprimer un besoin au recruteur qui se charge de lui trouver le bon profil. Le candidat n'a jamais de relations avec elle. Seul le recruteur peut éventuellement lui transférer un lien vers une séance d'entretien.

Les relations entre les 3 figurants sont résumées dans le schéma suivant :



I. Rapport technique

1. Cahier des charges

Il nous a été fourni au début du projet un cahier des charges précis à respecter.

a. Fonctionnalités

Pour le recruteur :

I) Gérer des questions

I.1 Enregistrer des vidéos

S'enregistrer facilement avec une webcam, directement sur le site internet.

I.2 Arrêter l'enregistrement

Arrêter l'enregistrement simplement sans avoir à spécifier la durée d'enregistrement au début.

I.3 Assurer le stockage et la persistance des données sur le serveur

Les vidéos doivent être stockées sur un serveur sans limite de durée et être accessibles par le recruteur.

I.4 Editer les propriétés des vidéos: sélectionner une durée maximale de réponse et un nom

Le but est de choisir à la fin de l'enregistrement un nom et une durée maximale de réponse pour la vidéo.

I.5 Visualiser les questions existantes

Visualiser une question après l'enregistrement, pour vérifier qu'elle est exploitable par exemple. Il faut aussi pouvoir visualiser les réponses des candidats.

II) Planifier des entretiens

II.1 Réaliser un entretien

Un entretien est composé de vidéos rangées dans un ordre précis. Il comporte aussi un nom (chef de chantier par exemple) et une liste de candidat auxquels il est envoyé.

II.2 Sélectionner des questions posées

Les candidats qui reçoivent un même recrutement répondent aux mêmes questions, dans le même ordre.

II.4 Sélectionner un ordre de réponse pour les questions

Les candidats répondent à la question juste après l'avoir visionnée.

II.5 Générer un lien par le biais d'une URL vers les vidéos

Le candidat ne reçoit pas les vidéos « en dur » dans le mail. Il reçoit un mail qui contient un lien vers une page du site web via laquelle il pourra répondre à l'entretien.

II.6 Envoyer un lien par courriel

La page d'envoi du mail doit être intégrée au site.

II.7 Etre notifié d'une réponse d'un candidat

Une fois une réponse reçue, le recruteur doit être informé et pouvoir visualiser la réponse en suivant le lien fourni.

III) Fonctions relatives à l'interface

III.1 Se connecter à un site en tant que recruteur

La connexion se fait via un login et un mot de passe.

III.2 Consulter la liste des personnes invitées à répondre

Consulter la liste des personnes qui ont reçu une invitation à répondre à un entretien.

III.3 Visionner les entretiens des personnes ayant répondues

Le recruteur doit pouvoir visionner les vidéos des réponses des candidats.

Pour le candidat :

I) Fonctions relatives à la connexion

I.1 Se connecter au site en suivant le lien reçu

La connexion se fait simplement en suivant le lien, aucun mot de passe ou login n'est nécessaire.

I.2 Passer par une phase de test pour vérifier sa connexion internet, sa webcam et son micro

La phase de test consiste à vérifier que la webcam et le micro du candidat sont compatibles avec l'application. La phase de test inclut également un test du débit internet.

II) Fonctions relatives à l'enregistrement/lecture des vidéos

II.1 Lire les vidéos du recruteur

Le candidat doit pouvoir simplement lire les vidéos du recruteur grâce à une application intégrée à la page internet.

II.2 Etre filmé dès que la question du recruteur est finie

La vidéo qui est enregistrée constitue la réponse du candidat. L'enregistrement de la réponse démarre automatiquement, dès la fin de la lecture.

II.3 Visualiser le temps restant pour répondre

Le candidat doit pouvoir, à travers un moyen simple, voir le temps qu'il lui reste pour répondre à la question.

II.4 Arrêt automatique du film

La webcam arrête automatiquement d'enregistrer si le temps fixé par le recruteur est dépassé ou si le candidat estime que sa réponse est finie et appuie sur « stop ».

III) Fonction gérant la fin de l'entretien

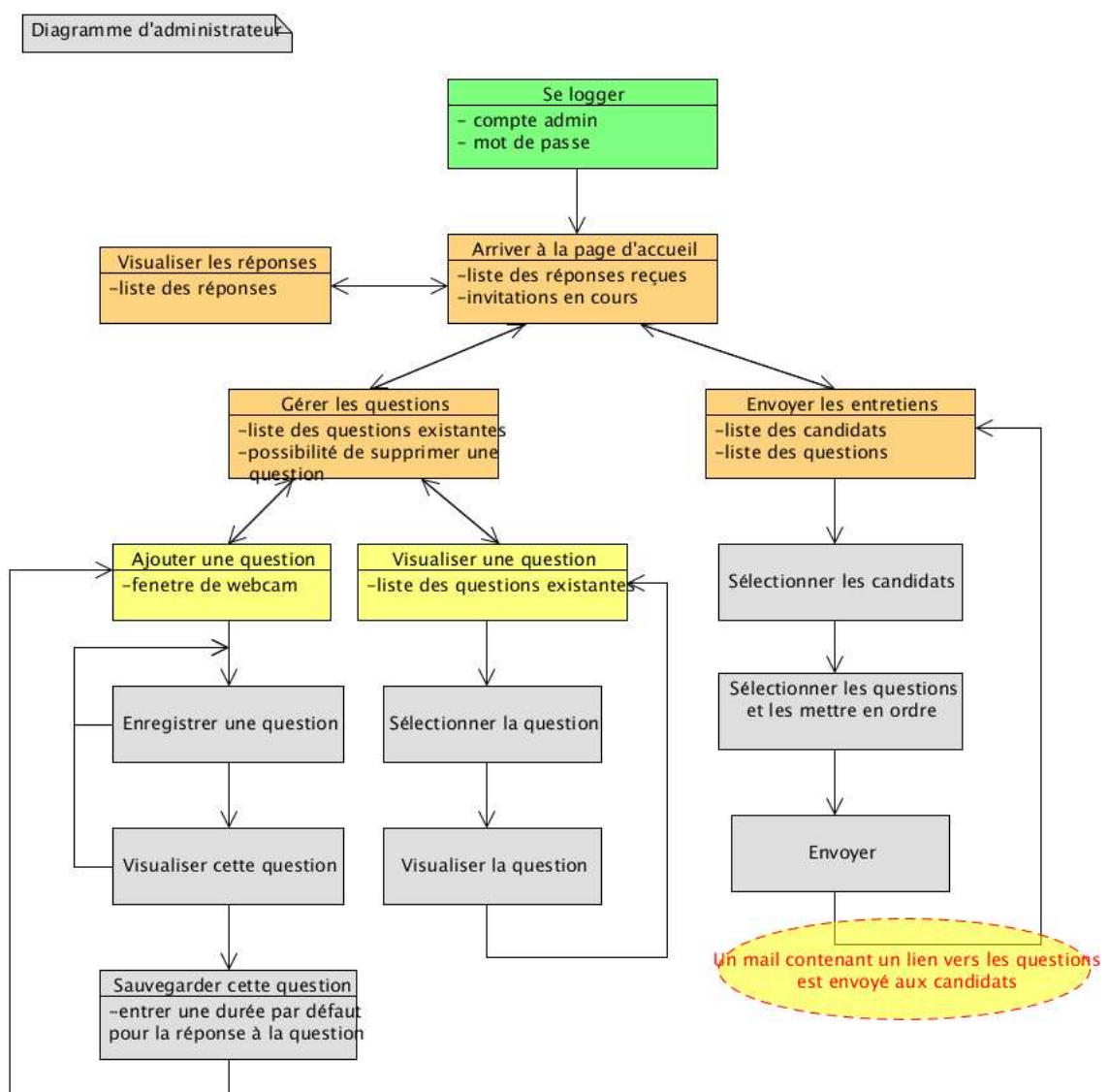
III.1 Arrêter la session une fois que le candidat a répondu à toutes les questions

L'entretien doit se terminer une fois que le candidat a répondu à toutes les questions de l'entretien.

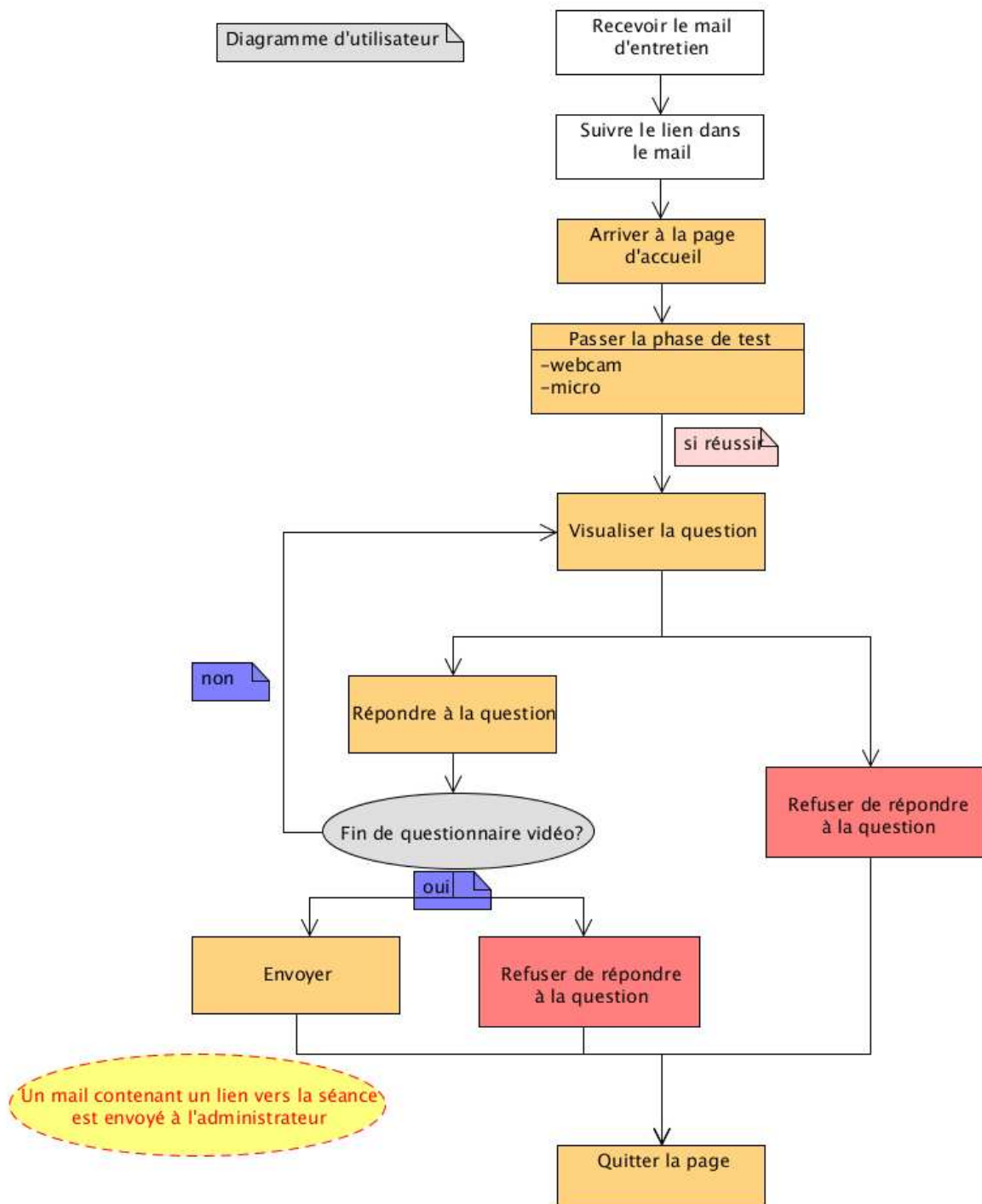
Enfin, au niveau des technologies utilisées, il nous a été imposé d'utiliser des pages jsp (issues du langage J2EE) pour le site web, ceci pour des raisons d'intégration de l'application dans une autre application plus globale, déjà écrite en J2EE.

b. Schéma global de fonctionnement (implémentation)

Le schéma global de fonctionnement pour le recruteur doit être le suivant :



Le schéma global de fonctionnement pour le candidat doit être le suivant :

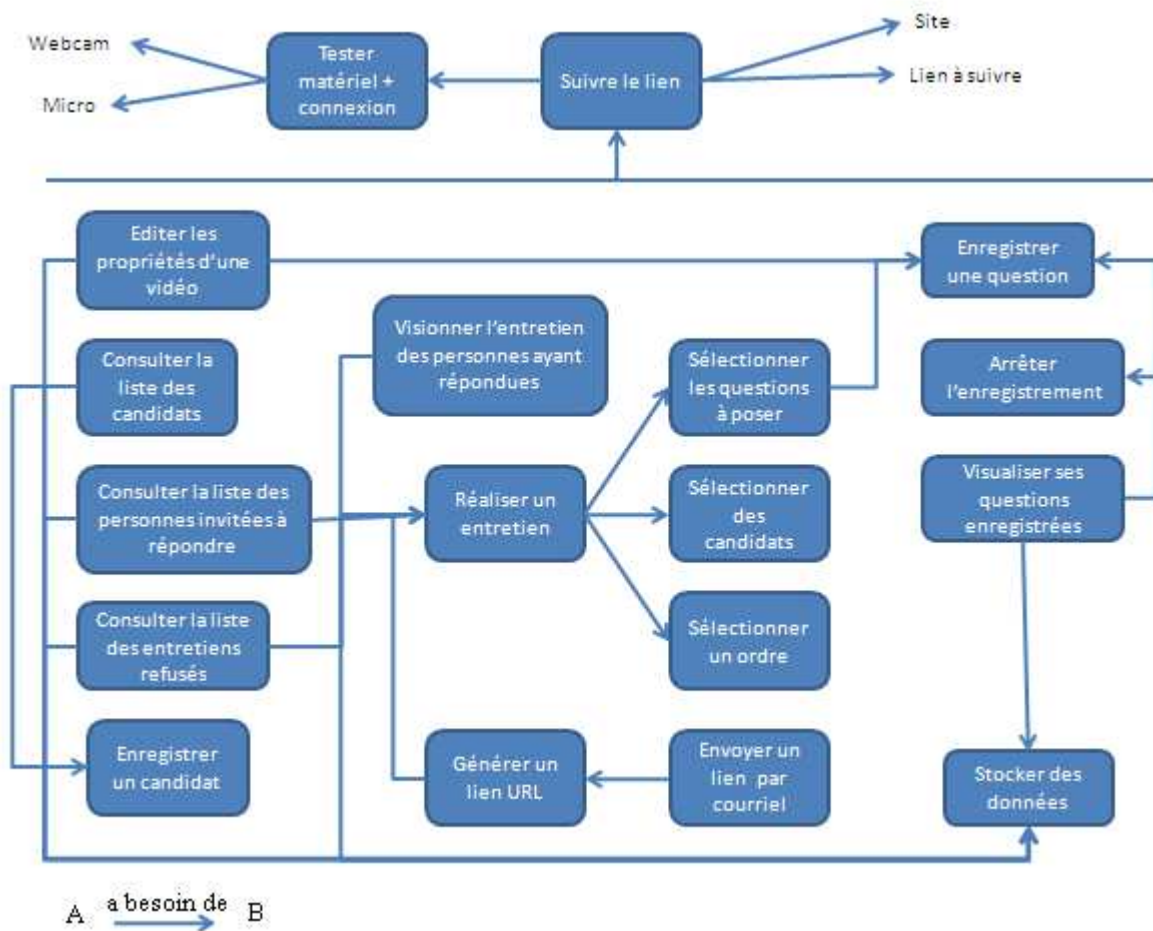


Nous allons présenter dans la partie suivante le choix des technologies utilisées ainsi que la solution que nous avons proposé pour répondre au besoin.

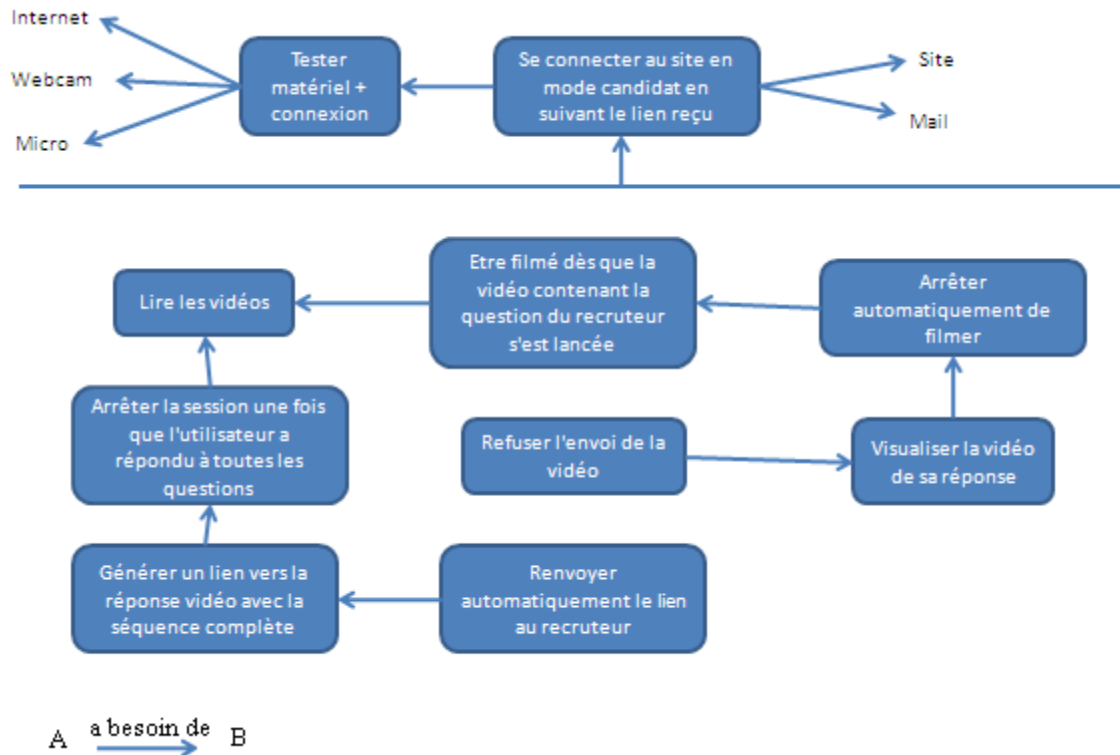
2. Spécifications fonctionnelles et détaillées

a. Dépendance entre les fonctions

Le schéma suivant présente les relations de dépendance entre les différentes fonctions pour le recruteur. Nous avons tenu compte en partie de ce schéma afin de ne pas perdre de temps.



Nous avons aussi du tenir compte du schéma suivant pour établir l'ordre de priorité concernant la réalisation des fonctions pour le candidat.



A noter qu'au final la fonction de refus d'envoi n'a pas été implémentée suite à une concertation avec l'entreprise.

b. Matériels et choix des technologies

Une bonne méthode pour justifier un choix étant souvent de fournir une analyse comparée des possibilités, nous présenterons donc nos choix d'algorithmes, de librairies ou de technologie (langage, serveur...) sous forme de tableaux comparatifs.

Mis à part le fait que le site web devait être réalisé en J2EE avec un serveur Tomcat et une base de données MySQL, aucune technologie n'était réellement imposée. Nous avons dû tester plusieurs technologies avant de trouver celles que nous avons choisies pour gérer l'enregistrement et la lecture vidéo.

Lire/enregistrer des vidéos

Nous avons sélectionné 2 langages : Flash et HTML5 et une bibliothèque multimédia Java : JMF. Il existe beaucoup d'API gérant le son et la vidéo en Java mais JMF est l'une des plus connues.

Nous présentons les avantages et inconvénients de chaque technologie dans le tableau suivant :

	Flash sur serveur RED5	HTML5	JMF avec applet
Avantages	+ streaming vidéo + open source + gère le temps	+probablement bien supporté par les futurs navigateurs internet	+ bien documenté + lire/écrire des vidéos depuis une applet + gère le temps + lire, capturer, sauvegarder, transmettre, transcoder
Inconvénients	- peu de tutoriels/aide disponibles - Supplantation du Flash par le HTML5 ? - plugin et technologie nécessitant des mises à jour constantes	- encore en développement - peu de documentation disponible	- installation nécessaire sur les postes client - peu de mises à jour - relativement peu de formats supportés

Au final, nous avons choisi la librairie Java JMF.

En effet, JMF est beaucoup utilisée et beaucoup de ressources traitent de ce sujet sur internet. Le projet ayant beaucoup de fonctionnalités à implémenter, il nous a paru important de commencer rapidement et de pouvoir bénéficier d'aide sur internet.

Le fait que la librairie ne supporte que peu de formats ne nous paraît pas rédhibitoire dans le cas de notre projet car il nous suffit d'en choisir un en particulier pour enregistrer/lire les vidéos. Nous n'avons besoin que d'un seul format.

Les divers types traités par JMF sont :

- protocoles: FILE, HTTP, FTP, RTP
- audio: AIFF, AU, AVI, GSM, MIDI, MP2, MP3, QT, RMF, WA
- vidéo: AVI, MPEG-1, QT, H.261, H.263, MOV

Nous avons choisi le format avi, qui est un format renommé et commun.

Choix du serveur

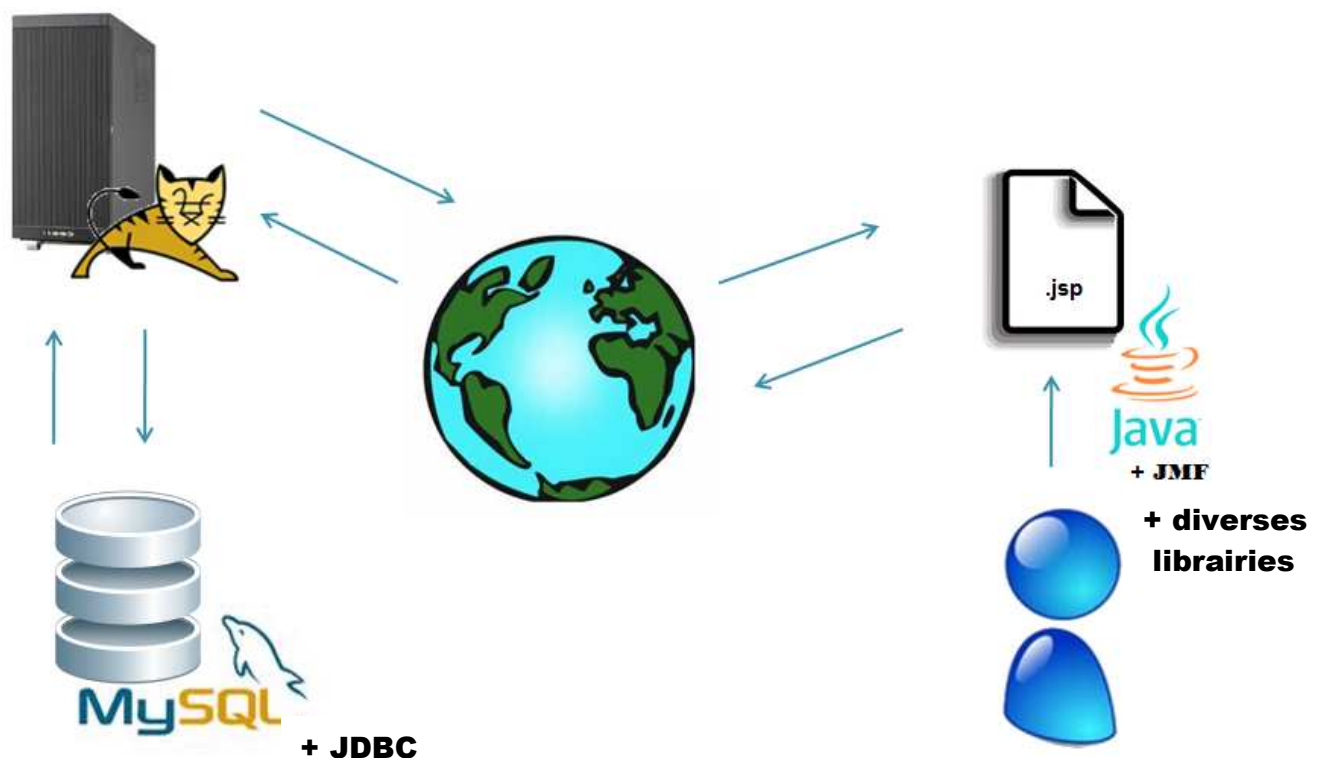
Etant donné que le projet doit être réalisé en J2EE, nous avons choisi d'utiliser un serveur Apache Tomcat. C'est en effet l'un des plus connus et de nombreux tutoriels sont disponibles. De plus, c'est celui que l'entreprise utilise et l'école en possède déjà un.

Base de données

La base de données permet de stocker, classer les données et de les identifier facilement.

Pour ce projet, nous avons choisi une base MySQL qui nous donne accès à nos tables. MySQL est un logiciel gratuit et open source. Il est fiable et fonctionnel tout en restant simple d'utilisation et d'administration. Nous avons prévu de l'utiliser avec le framework JDBC.

Au final, notre application s'articule de la manière suivante :



Le candidat (en bleu) accède aux pages jsp du site internet. Celles-ci contiennent des applets Java réalisées avec la bibliothèque JMF, qui nous servent pour l'enregistrement / lecture vidéo. Les pages jsp sont décryptées par le serveur Tomcat en place dans les locaux de Beetween. Ce dernier interagit également avec la base de données MySQL pour pouvoir assurer la persistance des données sur le serveur.

A noter: Nous ne stockons que les urls des vidéos dans la base de données.

c. Spécifications fonctionnelles

Comme nous l'avons vu, nous avons choisi de développer les applications à partir d'applet. Ce choix nous a paru judicieux puisqu'une applet est faite en Java (il est donc possible d'utiliser JMF) et est fait pour s'intégrer dans une page internet.

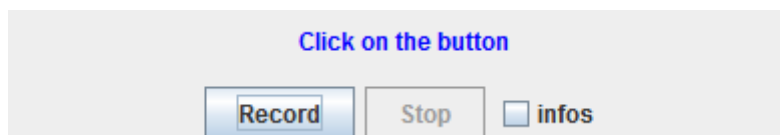
Nous présentons ici les applets de base de l'application.

Applet de test

Le recruteur et le candidat peuvent tester le fonctionnement de leur webcam en lançant cette applet. Lorsque l'image s'affiche, il est demandé de vérifier que la caméra est bien placée et que la luminosité est correcte.



Applet d'enregistrement

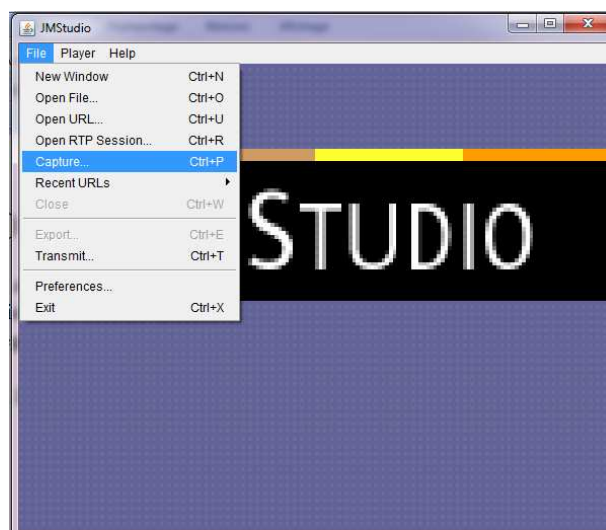


L'applet d'enregistrement permet d'enregistrer une vidéo.

Le label bleu affiche des informations sur les périphériques détectés (audio et vidéo) ainsi que sur les formats supportés. Il est donc possible de savoir si notre matériel est compatible (il faut pour cela cocher la checkbox « infos »). On peut arrêter l'enregistrement quand on le souhaite en cliquant sur « stop ». Le chemin d'enregistrement de la vidéo est la racine du disque Windows : C:\ et le format d'enregistrement est le .avi.

L'applet d'enregistrement qu'utilise le candidat est légèrement différente puisqu'il n'a pas à cliquer sur le bouton « Record ». En effet, elle prend des paramètres en argument. On lui passe notamment le chemin de la vidéo à visionner, le nombre total de vidéos etc...

Ayant rencontré des problèmes de fonctionnement sous Linux (que nous détaillerons dans la partie test), nous avons cherché une alternative pour ce système d'exploitation. Il suffit en effet de lancer une commande dans le terminal et d'utiliser le programme JmStudio qui est installé en même temps que JMF. Ce programme sert à tester le streaming et l'enregistrement vidéo. Avec celui-ci et la commande ajoutée, l'enregistrement fonctionne.



Applet de lecture



L'applet de lecture permet au recruteur de visualiser les questions qu'il a enregistrées ou les questions que les candidats lui ont renvoyées. Il peut choisir via l'onglet « Open » la vidéo qu'il souhaite lire.

Le candidat a une applet légèrement différente puisqu'il ne peut pas choisir le fichier qu'il souhaite lire. La page jsp lui transmet directement les vidéos à lire ainsi que diverses autres informations en argument.

Cette applet peut lire les vidéos aux formats avi, mpg et mov et accepte les vidéos qui n'ont pas été enregistrées avec l'applet d'enregistrement.

[La communication entre pages jsp](#)

Le J2EE est un langage relativement complexe mais très puissant.

Afin de créer une application dynamique, nous avons utilisé des méthodes pour communiquer et transmettre des informations entre les pages jsp.

Expliquons ceci sur un exemple.

Quand le candidat reçoit un entretien, le nom de l'entretien et son nom de famille sont transmis via l'url:



Une fois que le candidat s'est connecté à cette url, la page index.jsp récupère les arguments et les met en session:

```
//mise en session du nom du recrutement
String nom_recrutement = request.getParameter("Param1");
session.setAttribute("nom_recrutement", nom_recrutement);
```

A partir de cet instant, toutes les pages qui seront appelées par la suite pourront récupérer le nom du recrutement en faisant

```
//récupération du nom du recrutement
nom_recrutement = (String)session.getAttribute("nom_recrutement");
```

Grâce au nom du recrutement, la page home.jsp peut avoir accès à toutes les autres

informations nécessaires (url, durée des vidéos, nombre de questions...) en interrogeant la base de données.

Une fois toutes les informations récupérées, le visiteur peut cliquer sur le lien menant à la page intermédiaire.jsp. Celle-ci appelle et transmet des infos à la page synchro_lecture.jsp via un objet request:

```
request.setAttribute("cpt", "0");  
request.getRequestDispatcher("synchro_lecture.jsp").forward(request, response);
```

L'objet request ne transmet des informations qu'à la page passée en argument. C'est notamment en cela qu'il est différent de l'objet session.

La page synchro_lecture se charge, à partir du compteur (initialisé à 0 et qui représente le numéro de la vidéo courante), de lire la vidéo correspondante. Elle passe pour cela l'url de la vidéo en argument à l'applet de lecture:

```
<jsp:plugin type="applet" code="PlayerAppletCandidat2.class" codebase="."/  
jreversion = "1.6" archive="PlayerAppletCandidat2.jar" width="480"  
height="480" >  
  
    <jsp:params>  
        <jsp:param name="url" value="<%=url%>" />  
    </jsp:params>  
  
</jsp:plugin>
```

L'applet récupère le paramètre via la méthode `getParameter()` et peut lire la vidéo.

Une fois la vidéo lue, le candidat est automatiquement redirigé vers la page synchro_enregistrement grâce à la requête `getAppletContext().showDocument(url)`; Cette page va enregistrer la réponse du candidat.

Quand c'est fini, le compteur pour la question courante est incrémenté et on est renvoyé à la page synchro_lecture.jsp où le schéma se répète.

[L'envoi de mail avec génération de lien](#)

Pour envoyer un mail, nous avons dû créer un compte Gmail et le configurer. En se rendant dans les paramètres des messages, il faut aller dans l'onglet "Transfert et POP/IMAP" puis activer POP ET IMAP. Il suffit ensuite d'enregistrer et le compte est configuré.

Pour pouvoir utiliser les fonctions d'envoi, il faut inclure une librairie externe (activation et mail.jar)

3. Tests et validation

Avant de tester l'ensemble de l'application, nous avons dans un premier temps codé et testé chaque fonction pour savoir si elles fonctionnaient séparément. Nous les avons ensuite réunies en les assemblant étapes par étapes pour construire l'application finale.

Nous avons commencé par le codage des fonctions relatives au recruteur. La raison à cela est que le recruteur nécessite plus de fonctions et que celles-ci peuvent être réutilisées pour le candidat. Nous avons dans un premier temps essayé d'enregistrer des vidéos et de les lire. Ensuite, nous avons essayé de les uploader sur un serveur distant. Le serveur distant était en fait un autre ordinateur connecté au nôtre par le routeur d'une box ADSL. Nous avons testé le réseau en local (avec des adresses types 192.168.0.3) et l'application fonctionnait.

Au niveau des tests de compatibilité, nous avons découvert que l'application était compatible avec les caméras intégrées dans nos deux ordinateurs portables sous Windows XP ainsi qu'avec une webcam de marque Labtec qui ne figure pas dans la liste fournie par Oracle. L'applet d'enregistrement ne fonctionne pas correctement sous Linux. Nous avons eu des problèmes avec le fait que JMF utilise une vieille version d'une librairie vidéo (video4linux). Nous pouvons résoudre en partie ce problème en inscrivant une commande dans le bash et en utilisant JmStudio. Etant donnée que l'immense majorité des gens utilisent Windows comme système d'exploitation, nous n'avons pas cherché à modifier plus que ça notre applet. Le recruteur qui travaille sous Linux pourra utiliser l'alternative décrite précédemment. Nous n'avons jamais de trouvé de problèmes au niveau des périphériques audios. Quand à l'applet de lecture, nous avons réussi à lire un film d'une heure trente au format mpg. Elle semble donc stable.

La validation des tests s'est faite sur notre avis, sur celui des testeurs et sur le respect du cahier des charges. Nous présentons dans la partie suivante un tableau récapitulatif.

4. Recette

Afin de prouver le fonctionnement de notre application, nous allons :

- jouer le scénario type d'un recruteur qui envoie un recrutement à un candidat. Le candidat reçoit les vidéos, les lit, et y répond.
- expliquer les limites de l'application

Le tableau suivant récapitule ce qui a été fait.

Fonctions	Etat	Commentaires
RECRUTEUR		
Gérer des questions		
Enregistrer des vidéos		
Arrêter l'enregistrement		
Visualiser les questions		
Base de données		
Propriétés des vidéos		
Planifier des entretiens		
Réaliser un entretien		
Sélectionner des questions		
Sélectionner un ordre		
Génération d'un lien		
Envoyer un mail		
Etre notifié d'une réponse		Page prévue mais reste à mettre en relation avec BDD
Planifier des entretiens		
Login / mot de passe		
Liste des personnes invitées		Manque de temps mais possible de le faire avec l'application actuelle
Visionner les entretiens		Il faut regarder dans le dossier si des nouvelles vidéos sont arrivées...
CANDIDAT		
Connexion		
Suivre le lien		
Phase de test		
Enregistrement / lecture		
Lire les vidéos recruteur		
Début du film après lecture		
Visualisation du temps		
Arrêt automatique ou planifié		
Fin de l'entretien		
Arrêt de la session		

Au final, nous estimons que 90% des fonctionnalités de base ont été implémentées.

Nous n'avons pas eu le temps de faire les améliorations.

A la question « est-ce que cette version pourrait être mise en production aujourd'hui ? », la réponse est « pas tout à fait ». Nous estimons à 70% le pourcentage de finalité du projet avant que celui-ci puisse être mis en ligne. En effet, il faudrait améliorer l'interface recruteur et trouver un moyen d'envoyer plus facilement les vidéos pour le candidat. Actuellement, celui-ci doit les chercher sur son disque dur et les entrer dans le champ « input » d'un formulaire HTML. Il est impossible de définir une valeur de fichier par défaut dans les formulaires HTML de type « input » pour raisons de sécurité. C'est pour cette raison que le candidat doit chercher les fichiers un à un.

A noter qu'il nous paraît tout à fait possible de finaliser entièrement le projet avec les technologies que nous avons utilisées.

5. Performances

Nous souhaitons avoir une application complète qui puisse tourner sur des ordinateurs ayant des configurations minimalistes (EeePc par exemple) sans pour autant qu'il ne consomme trop de mémoire ou de temps processeur.

En plein enregistrement ou lecture vidéo (qui nécessite beaucoup de ressources), le programme consomme entre 15 et 25% du CPU et 40 Mo de mémoire (sur un processeur Intel Atom @ 1.6GHz) ce qui nous semble convenable pour ce projet.

6. Difficultés techniques rencontrées et solutions apportées

Enregistrement de vidéos sur des ordinateurs ayant des périphériques différents

C'était véritablement la plus grosse difficulté technique. Nous avons trouvé quelques codes d'applications java avec JMF qui permettent d'enregistrer des vidéos mais toutes ou presque étaient faites pour un type précis de webcam. Or, étant donné que nous avons besoin d'un programme qui fonctionne avec un maximum de webcams, la seule solution que nous avons trouvée est de faire une liste des périphériques audio et vidéo trouvés et d'essayer avec chacun d'entre eux jusqu'à ce qu'il y en ait un qui fonctionne.

Nous avons aussi dû gérer les problèmes de formats vidéo et de codecs. En effet, il a fallu

s'assurer que le format dans lequel l'applet d'enregistrement enregistre les vidéos est un format lisible par l'applet de lecture. Après plusieurs tests, nous avons choisi le format avi.

Ce format a aussi l'avantage d'être relativement générique et donc de pouvoir être lu et enregistré sur d'autres lecteur / enregistreur que le nôtre.

En plus de ces nombreux problèmes d'encodage des vidéos, il nous a fallu nous assurer de créer une application qui ne soit pas de type « usine à gaz » car la vidéo consomme beaucoup de ressources CPU.

Streaming vidéo avec JMF

Nous avons pensé à utiliser le streaming vidéo que permet JMF. Cela aurait permis d'éviter au candidat d'uploader un par un les fichiers qui constituent sa réponse.

Toutefois, après de nombreux tests, nous n'avons jamais réussi à faire fonctionner le streaming correctement, sauf en lecture.

Au niveau des applets

Une difficulté que nous avons sous-estimée concerne les applets Java. Ainsi, une fois l'application d'enregistrement construite en application java, il a fallu la transformer en applet, ce qui nécessite quelques modifications. Pour pouvoir intégrer l'applet à la page internet, il a fallu la transformer en archive jar contenant toutes les bibliothèques JMF additionnelles. Cela paraît simple mais nous avons eu des difficultés. Nous n'avons pas réussi à créer des fichiers manifest correct que nous transformions avec des lignes de commande. Au final, nous avons trouvé une librairie additionnelle appelée FatJar qui fait tout cela très simplement, via une interface graphique.

Enfin, il a fallu se renseigner et réfléchir à la façon dont nous pouvions transmettre des paramètres à une applet et en recevoir à notre tour.

Communications entre jsp pour voir un contenu cohérent

Cette difficulté se réfère plus largement à celle de devoir apprendre un nouveau langage seul, uniquement avec les ressources disponibles sur internet ou dans les livres, le tout dans un intervalle de temps limité.

7. Conclusion et perspectives

Finalement, nous avons une version « 0 » de l'application. La majorité des fonctionnalités de base ont été implémentées et fonctionnent correctement mais il reste quelques améliorations à faire pour aboutir véritablement à une version « 1 », utilisable et qui pourrait être mise en ligne.

Quelques améliorations pourraient être ajoutées :

- avoir la possibilité de regarder les questions et/ou les réponses dans les vidéos reçues par le recruteur.
- pouvoir accéder à une séquence précise de la vidéo, via des marque-pages par exemple
- partager automatiquement (via l'application) les vidéos avec une entreprise ou un autre recruteur
- recevoir des commentaires de l'entreprise sur une vidéo que le recruteur a envoyée
- visualiser les questions en mode texte 'pour les malentendants.

II- Rapport de gestion du projet

Introduction

La gestion de projet est une partie indispensable au bon déroulement de celui-ci. Les compétences techniques de chacun sont à prendre en compte mais une gestion de projet correcte est aussi capitale pour mener le projet à bien le plus efficacement possible.

1. Découpage en tâches

Afin de mener à bien notre projet, il nous a fallu déterminer les différentes tâches, les découper et les répartir équitablement entre nous.

Pour cela, nous avons, pendant les quatre premières semaines, pris connaissance du sujet, établi les fonctions nécessaires pour remplir les besoins, fait des recherches sur les technologies que nous pouvions utiliser et déterminé les tâches à accomplir.

Notre projet est composé de 2 parties essentielles : la construction du site internet et l'enregistrement / lecture des vidéos.

Les cinq principales tâches que nous avons déterminé et qui vont être la base du projet sont les suivantes:

- Enregistrement / Lecture des vidéos
- Gestion de la base de données
- Construction du site internet
- Phase de test pour le candidat
- Connexion au serveur mail et envoyer des courriels

2. Organisation de l'équipe

L'organisation de l'équipe est une chose importante pour le bon fonctionnement du projet et est faite selon nos capacités, nos connaissances et nos personnalités.

Initialement, sachant qu'Antoine avait déjà travaillé en HTML/CSS, il a été chargé de la construction du site internet et de l'envoi d'email. Huiting devait s'occuper de l'enregistrement et de la lecture de vidéos, ainsi que de la création de la base des données. Etant donné que la phase de test concerne la lecture / l'enregistrement et la structure du site, nous avons décidé de la faire ensemble.

Cependant, au bout de 3 semaines, nous avons eu un problème avec JMF sous Linux. Pour continuer à avancer, nous avons échangé les tâches.

Nous avons travaillé en relative autonomie sur ses tâches en expliquant clairement chaque modification effectuée et l'utilisation des fonctions réalisées. Ainsi, chaque mardi après-midi nous avons réunis nos fichiers et ajouté nos modifications afin que nous ayons chacun une version correcte et complète pour travailler pendant la semaine.

Etant donné que nous n'étions que deux, nous travaillions beaucoup par échange de fichiers sur clé USB ou par mails en expliquant les modifications effectuées.

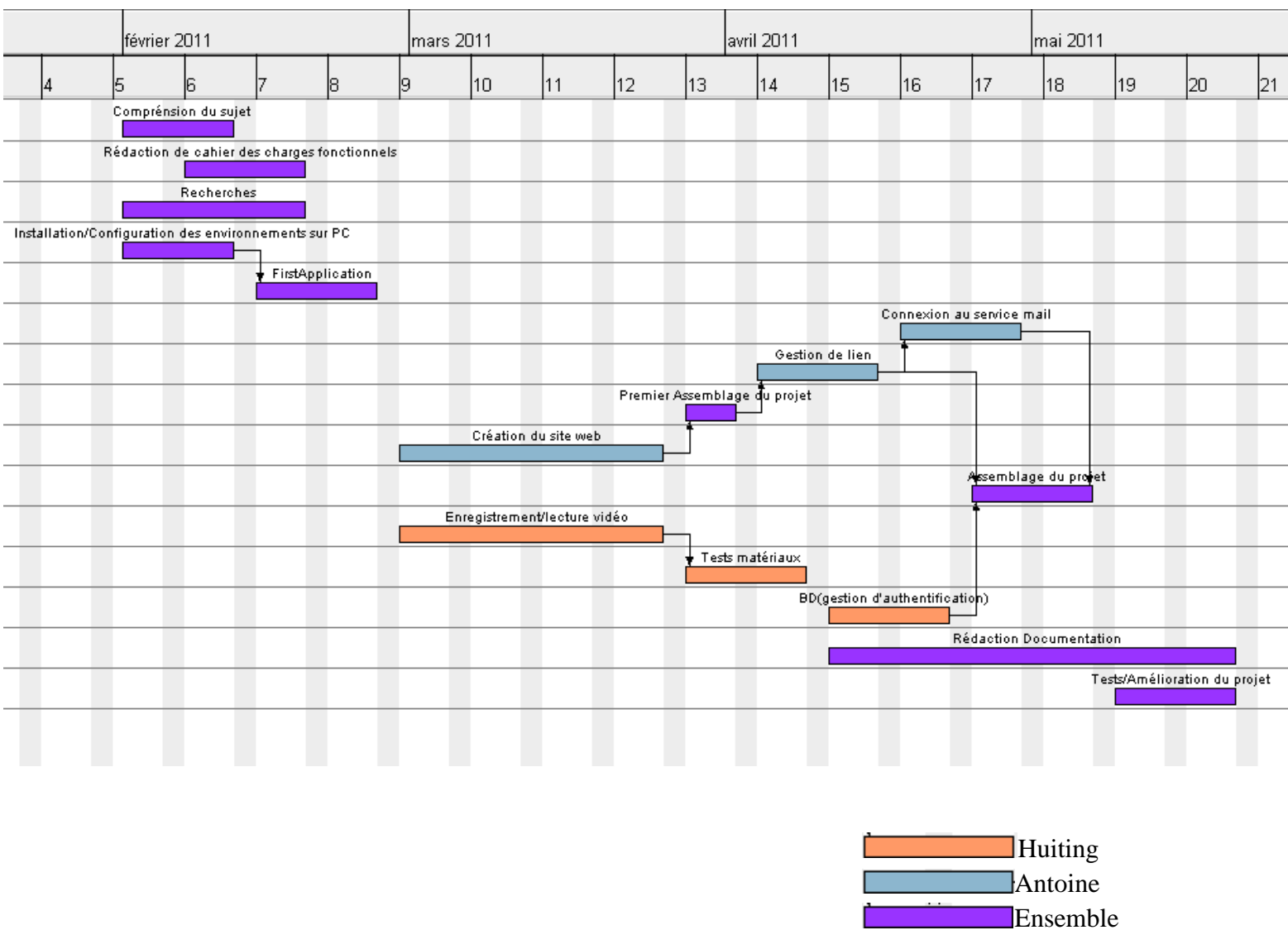
Afin de maintenir la cohésion au sein de l'équipe, chaque semaine nous faisons le point sur nos tâches en début de séance, et si nous avons des idées nous pouvions donner notre point de vue.

Ces mini-réunions ont évité la dispersion des idées et chacun a pu s'exprimer librement et contribuer efficacement à l'amélioration du projet.

Au début et à la mi-projet,, nous avons rencontrés M. Muset, de l'entreprise *Beetween* pour discuter et confirmer nos solutions proposées. Ensuite, nous avons fait un rapport par semaine pour l'entreprise. Le but de ce rapport par mail était de rendre compte l'avancement du projet et des perspectives d'avancement pour la semaine suivante.

3. Diagramme de Gantt initial

La répartition et l'organisation ont abouti à la réalisation de ce premier diagramme de Gantt établi à l'aide du logiciel *GanttProject*.



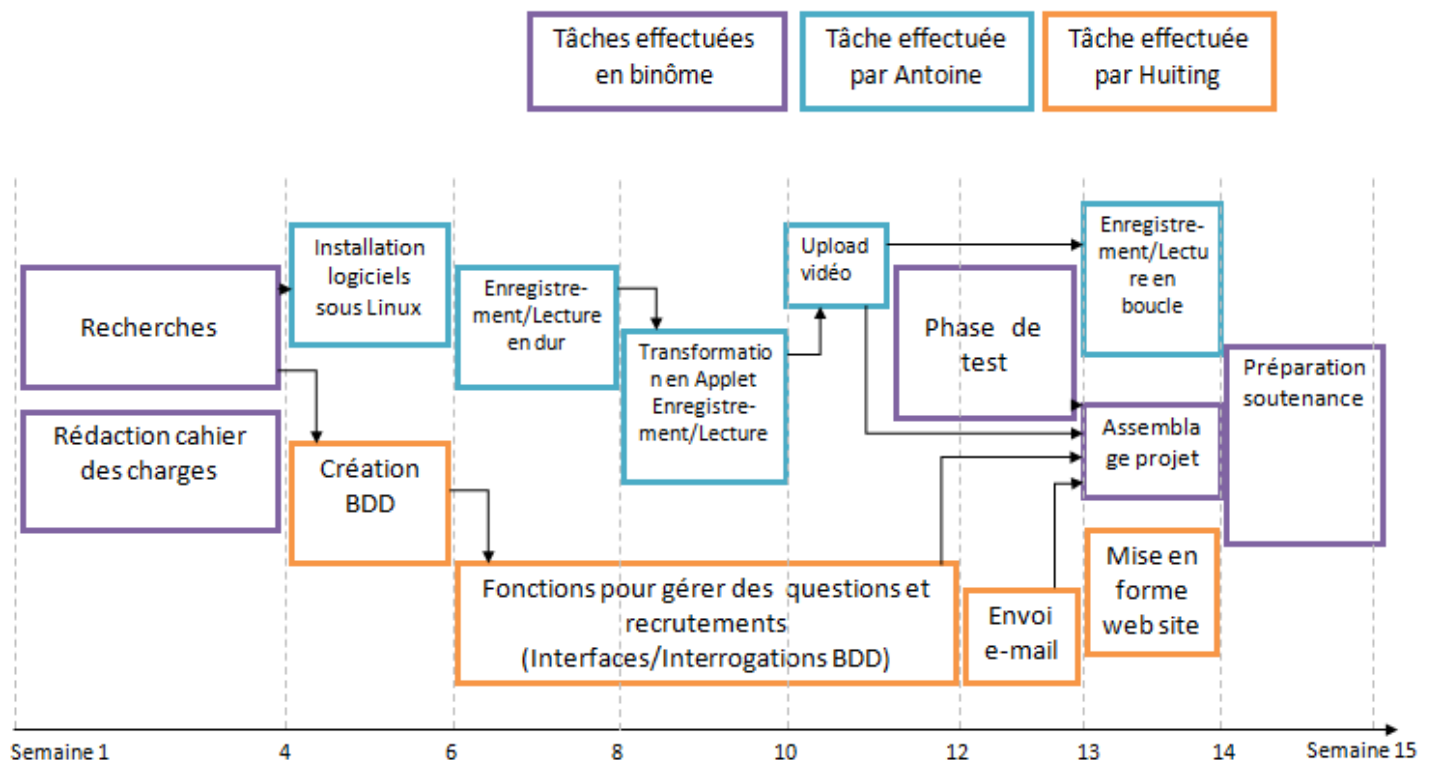
4. Difficultés rencontrées dans la gestion du projet et solutions apportées

Malgré cette organisation, il y a eu quelques petits soucis pour déterminer la durée de certaines tâches. En effet, comme nous n'avions jamais travaillé dans le domaine des technologies vidéo avec jsp auparavant, il nous a paru difficile d'évaluer le temps nécessaire pour chaque tâche. Certaines fonctions sont fortement reliées entre elles mais elles sont effectuées séparément par chacun de nous. Il fallait alors mettre en commun ce qui a causé quelques problèmes. Néanmoins, en travaillant ensemble, nous avons pu trouver une méthode de travail efficace et résoudre les problèmes.

Nous n'avons pas eu de problèmes d'entente.

5. Diagramme de Gantt réalisé

Finalement, le diagramme de Gantt initial n'a pas tout à fait été respecté, beaucoup de nouvelles tâches sont apparues et certaines tâches nous ont pris plus de temps que prévu.



6. Conclusion et perspectives

Au final, nous avons beaucoup travaillé pendant les séances et hors des séances. Nous avons compris qu'il est important d'évaluer les connaissances de chacun, et de faire un point sur nos tâches régulièrement.

Notre organisation était plutôt bonne. Il y avait une bonne cohésion de groupe. Nous avons appris à gérer un projet et à faire face aux difficultés ensemble.

III- Apport du projet

Ce projet long de 15 semaines est une bonne expérience, il nous a apporté beaucoup, tant au niveau technique qu'en terme de gestion de projet.

1- Apports scientifiques et techniques

Ce projet nous a permis de découvrir le monde de la vidéo sur internet, un monde complexe mais passionnant de par les débouchés auxquels il mène. Nous avons également acquis des notions sur un nouveau langage orienté web et découvert beaucoup de chose sur un domaine qui nous était totalement inconnu : celui des serveurs. De plus, nous avons créé un site qui répond à une vraie problématique issue du monde de l'entreprise, ce qui est très gratifiant. Les acquis de ce projet nous serviront dans le futur.

2- Apport sur la gestion de projet

Nous avons confirmé le fait que la communication est primordiale lorsque l'on travaille ensemble. Un dialogue par mail ou messagerie instantanée ne remplacera jamais une entrevue en face à face.

Il faut toujours réussir à motiver l'autre par les idées que l'on apporte et réfléchir avant de se

lancer dans une voie.

3- Apport sur la formation pédagogique

Ce projet nous a permis de consolider nos connaissances sur le langage JAVA/J2EE et MySQL. Nous avons appris à utiliser une bibliothèque multimédia (JMF) ainsi que diverses autres librairies plus petites mais pas moins inutiles (mail, upload...)

Ce projet constitue notre première création de site web pour une entreprise. Nous sommes satisfaits d'avoir choisi ce sujet car le domaine de l'internet est très peu développé dans notre formation.

4- Apport personnel

Chaque membre de l'équipe a eu des apports personnels durant le projet.

Antoine :

Expérience intéressante qui lui a permis de coréaliser un projet abouti et qui lui a appris à être plus à l'écoute des autres. Que du positif, expérience à renouveler !

Huiting :

Sujet intéressant qui lui a donné envie de travailler. La bonne communication a facilité sa compréhension.

Conclusion

C'est la première fois que nous travaillons en groupe sur un projet avec un but bien défini. De l'avis général, nous avons consolidé nos connaissances générales et appris à faire des applications plus attrayantes et plus orientées pour le monde du travail. Nous sommes globalement satisfaits de ce que nous avons réalisé.

Au niveau de la gestion du projet en équipe, nous avons réussi à bien nous répartir les tâches afin de réaliser nos objectifs dans les temps et l'ambiance générale du groupe était très bonne.

Une bonne expérience à renouveler !

ANNEXES

Menu

- [Home](#)
- [Créer un recrutement](#)
- [Enregistrer une question](#)
- [Envoyer des fichiers](#)
- [Visualiser une question](#)
- [Visualiser ma webcam](#)
- [Supprimer une question](#)
- [Envoyer un courriel](#)
- [Envoyer un recrutement](#)
- [Liste vidéos](#)

Espace de travail



Ajouter un recrutement



Enregistrer une question



Envoyer des fichiers



Visualiser une question



Visualiser ma webcam



Supprimer une question



Envoyer un courriel



Envoyer un recrutement



Liste video

Copyright "Questionnaire Video" 2011, tous droits réservés

Menu

- [Home](#)
- [Créer un recrutement](#)
- [Enregistrer une question](#)
- [Envoyer des fichiers](#)
- [Visualiser une question](#)
- [Visualiser ma webcam](#)
- [Supprimer une question](#)
- [Envoyer un courriel](#)
- [Envoyer un recrutement](#)
- [Liste vidéos](#)

Lecture:



Menu

- [Home](#)

Menu

- [Home](#)
- [Créer un recrutement](#)
- [Enregistrer une question](#)
- [Envoyer des fichiers](#)
- [Visualiser une question](#)
- [Visualiser ma webcam](#)
- [Supprimer une question](#)
- [Envoyer un courriel](#)
- [Envoyer un recrutement](#)
- [Liste vidéos](#)

To:

Objet:

Body:

Vous avez un entretien vidéo avec Beetween, veuillez vous connecter dès que possible à cette adresse:

http://localhost:8080/x_projet/file_candidat/index.jsp?Param1=Chef de chantier&Param2=Jean



Ajouter une question

Dossier de recrutement:

Nom de la question

Nom de la vidéo:

... choisi

Durée maximale pour la réponse:

min

sec

