

IUT2 INFO - Université Grenoble Alpes

M3301 - Itération 1  
Compte-Rendu Groupe 8

---

## **Logiciel de suivi de cours**

---

GUITON Clément  
CEREN Carlos  
DUPLESSIS Thomas  
FRAGNON Adrien  
LAROUSSE Abel  
VIAUD Nathan

## Table des matières :

<b>Cadrage du projet</b>	<b>2</b>
Contexte et Objectifs	2
Analyse du terrain et de l'existant	2
Contraintes et Risques	3
<b>Expression des besoins</b>	<b>3</b>
Identification des besoins fonctionnels	3
Priorisation des besoins fonctionnels	3
Mise en avant des critères qualité logicielle	4
Mise en avant des critères ergonomiques	4
<b>Solution</b>	<b>4</b>
Solution proposée	4
Réduction de la bande passante	4
Annotation	4
Question des élèves	5
Quizz en direct	5
Présentation des solutions techniques	5
Solution générale	5
Interface IHM	5
Accueil	5
Vue lecteur	6
Vue présentateur	6
Gestion du travail	6
Discord	7
Trello	7
Tests	7
Répartition des tâches	7
Description d'un modèle de développement	7
<b>Annexe</b>	<b>9</b>
Annexe 1	9
Annexe 2	10
Annexe 3	11

# 1. Cadrage du projet

## 1.1. Contexte et Objectifs

Pendant l'année scolaire 2019-2020, le monde a fait face de manière spontanée à une pandémie mondiale qui a obligé le système éducatif français à prendre la décision d'annuler les cours en présentiel. Cet événement a obligé l'administration à prendre une décision au plus vite pour s'assurer que les étudiants continuent le suivi de leur filière. La solution choisie a été des cours en distanciel. Ceci était un réel défi pour les professeurs afin de trouver des logiciels adaptés à chaque enseignement. En effet ceux-ci devaient répondre à plusieurs critères : la lecture d'un fichier depuis la machine professeure jusqu'aux machines étudiantes ainsi que la communication écrite ou vocale entre les enseignants et les étudiants - pour ne citer que les plus importants.

Ainsi, plusieurs solutions ont été prises mais ayant chacune des défauts non négligeables. Dans l'initiative de notre projet, nous allons chercher comment est-il possible d'assurer les enseignements à distance tout en prenant en compte des conditions de connexion variables des concernés soit le corps enseignant et les étudiants.

## 1.2. Analyse du terrain et de l'existant

Lors du confinement, nous avons principalement utilisé Discord pour les cours à distance. Le principal problème était l'instabilité des connexions internet (côté professeur ou élève). Cependant, cette plateforme a un inconvénient majeur pour le suivi des cours : il diffuse un flux vidéo. Malheureusement, la connexion de certains étudiants et de certains enseignants n'était pas assez importante pour supporter ce flux ce qui pouvait provoquer des coupures, voire rendre le texte illisible à l'écran.

Certains enseignants utilisaient seulement Discord pour la communication vocale, nous devons donc suivre un PDF sur notre machine. Cependant il était parfois compliqué de suivre les cours : si nous voulions revenir en arrière sur le cours ou si nous avions tout simplement perdu notre attention pendant quelques instants, il devenait difficile de retrouver le fil du cours.

Un autre inconvénient des logiciels de chat en ligne est la capacité des élèves à poser des questions. En effet, pour cela les élèves doivent soit poser une question dans le chat qui risque de ne pas être remarqué dans le flot continu de texte, ou alors intervenir à l'oral et couper le professeur, ce qui peut déranger le déroulement du cours ou freiner les élèves les plus timides.

Nous avons pensé à Google Docs qui est un logiciel de traitement de texte collaboratif en ligne composé de curseurs, fonctionnalité que nous reprendrons, mais cet outil n'est pas vraiment adapté pour une présentation et ne permet pas le suivi du fil du cours.

Dans ces deux logiciels, il y a des difficultés à poser des questions. Le professeur ne peut pas poser de questions aux élèves sans qu'il y ait de trop nombreuses réponses dans le salon textuelle, rendant impossible à l'enseignant d'avoir un réel compte rendu des résultats. Nous avons ensuite découvert BigBlueButton, ce site ressemble beaucoup à l'idée de base que nous avons de notre application, cependant, BigBlueButton a le même problème que Discord : Il diffuse un flux vidéo.

De plus, plusieurs améliorations nous ont été proposées par des professeurs, comme par exemple la possibilité pour les professeurs de préparer les quizz à l'avance.

### 1.3. Contraintes et Risques

Ce projet doit respecter quelques contraintes. En effet, nous avons premièrement un délai limité pour réaliser notre projet. Nous avons aussi une limite technologique puisqu'un de nos buts est de fournir un service qui ne nécessite pas un débit trop élevé. Il faut aussi que l'application puisse tourner sur des ordinateurs peu puissants afin de s'adapter à tous les étudiants. Il y a aussi une contrainte de développement puisque nous sommes limités par les technologies présentes à l'IUT et à ce qu'elles ont à nous proposer. Une grande stabilité doit aussi être présente puisque le domaine traité ici est l'enseignement, un problème technique impactant l'apprentissage n'est ici pas envisageable.

Le projet comporte peu de risques, premièrement, un premier est lié à l'augmentation des informations puisque cela pourrait surcharger visuellement le client. Nous pouvons aussi craindre une augmentation de la bande passante nécessaire si trop de fonctionnalités futiles sont ajoutées.

## 2.Expression des besoins

### 2.1. Identification des besoins fonctionnels

Dans ce projet, nous trouvons plusieurs besoins fonctionnels. Ceux-ci doivent résoudre les problèmes mentionnés précédemment. On trouve par exemple le fait de pouvoir accéder au pdf sans avoir besoin d'une bonne connexion d'internet, ce qui pourra résoudre les problèmes de connexion présentés par des étudiants ou des professeurs. De plus, la possibilité des professeurs de pouvoir ajouter des notes dans le pdf de cours va être un besoin important dans notre application de suivi de cours. Ensuite, la possibilité de pouvoir télécharger à la fin du cours le pdf annoté sera une autre fonctionnalité sur laquelle nous nous pencherons. De même, la présence d'un bouton "poser une question" pour les étudiants sera envisagée par notre application web. Finalement, la possibilité de lancer des petits quizz type QCM par les enseignants est un autre besoin dans notre application. De ce fait, les professeurs peuvent savoir en temps réel si la notion traitée est bien comprise par les étudiants.

### 2.2. Priorisation des besoins fonctionnels

Tous les besoins énumérés dans le paragraphe précédent ont été priorisés. Premièrement, on peut remarquer que le plus important est la réduction de la bande passante. Nous pensons que celui-ci est le plus important car le plus grand problème rencontré pendant le confinement était le fait de ne pas pouvoir suivre les cours dû à des mauvaises connexions. Par la suite, les annotations pour les professeurs sont aussi très importantes. La cause étant que pendant les cours en distanciel, quand les professeurs expliquaient certaines notions, aucun moyen n'était présent afin de mettre ces annotations dans les slides du cours. Subséquemment, le bouton "lever la main" sera notre troisième besoin. Cela s'explique par le fait que dans les salons écrits des serveurs Discord dédiés aux cours, les messages des étudiants n'étaient parfois pas lus par les professeurs à cause de la rapidité de défilement des messages. Enfin, la fonctionnalité du quizz est identifiée

comme notre dernier besoin. En effet, nous pensons que les besoins mentionnés postérieurement sont plus importants que la mise en place d'un quizz.

## 2.3. Mise en avant des critères qualité logicielle

Nous avons choisi des critères qualités concernant notre projet qui nous tenait à coeur, pour commencer nous voulons principalement nous concentrer sur les critères de facilité d'usage comme la facilité d'apprentissage et de compréhension, nous voulons qu'il y ait le moins de temps de compréhension possible, que tout le monde puisse lancer l'application et l'utiliser sans problème. L'exploitabilité est également un critère important, il faut que l'utilisateur puisse sans investissement sérieux comprendre toutes les possibilités qui lui sont offertes. Enfin, la stabilité du système est primordiale afin de garantir une expérience optimale à l'utilisateur.

## 2.4. Mise en avant des critères ergonomiques

En ce qui concerne les critères ergonomiques nous voulons nous concentrer sur la distinction et le groupement des informations afin que l'utilisateur comprennent rapidement quel élément de l'interface est relié à quelle fonctionnalité. Nous souhaitons également insister sur la lisibilité et la diminution de la densité informationnelle afin que l'utilisateur ne se perde pas dans des données superflues et puissent interagir avec l'interface le plus efficacement possible.

# 3. Solution

## 3.1. Solution proposée

### 3.1.1. Réduction de la bande passante

Pour résoudre les problèmes évoqués précédemment, voici les solutions auxquelles nous avons pensé. Chaque utilisateur possède son propre lecteur pdf exécuté dans son navigateur. Un pdf contenant le cours est transmis à chacun et ensuite, lorsque le présentateur change de page, le numéro de celle-ci est transmis au serveur qui le renvoie à chaque "lecteur". Cette technique limite donc le transfert de données à de simples numéros de page (pour simplifier) au lieu de transmettre un flux vidéo complet qui est beaucoup plus gourmand en terme de bande passante.

### 3.1.2. Annotation

En plus de cette idée qui résout le problème de base d'inégalité de connexion, s'ajoutent d'autres fonctionnalités nécessaires à un enseignant pour faire cours. La première est la possibilité d'ajouter des commentaires sur le document. Ces commentaires sont transmis à chaque lecteur en direct, tous les utilisateurs ont la possibilité de télécharger le document contenant les annotations en fin de session.

### 3.1.3. Question des élèves

Une fonctionnalité qu'on trouve intéressante à ajouter est de donner la possibilité aux étudiants de poser une question à l'enseignant en envoyant une notification assez visible. Cela évitera de couper la parole au professeur ou que la question soit oubliée dans le chat.

### 3.1.4. Quizz en direct

Une autre fonctionnalité qui pourrait être utile non seulement pour les étudiants, mais également pour les enseignants, est la possibilité de pouvoir créer à l'avance des quizz qui pourront être lancés pendant le cours. Ainsi les résultats de ces petits QCM seront envoyés en direct aux enseignants, afin de voir si la notion abordée est bien assimilée par les étudiants, ou s'il est nécessaire de revenir dessus. De même les étudiants pourront voir s'ils ont bien compris ce qui a été expliqué. Les QCM peuvent être préparés à l'avance ou, si le professeur le voit nécessaire, être faits rapidement pendant le cours et le lancer.

## 3.2. Présentation des solutions techniques

### 3.2.1. Solution générale

Pour résoudre les problèmes évoqués précédemment, voici nos solutions techniques. Chaque utilisateur possède son propre lecteur pdf exécuté dans son navigateur grâce à un script javascript. Un pdf contenant le cours est transmis à chacun et ensuite, lorsque le présentateur change de page, le numéro de celle-ci est transmis au serveur qui le renvoie à chaque "lecteur". Cette technique limite donc le transfert de données à de simples numéros de page (pour simplifier) au lieu de transmettre un flux vidéo complet qui est beaucoup plus gourmand en terme de bande passante. Pour mettre en œuvre cette solution, plusieurs technologies sont nécessaires. Côté serveur, un serveur HTTP qui gère les connexion des utilisateurs, qui leur transmet les bonnes pages html ainsi qu'un serveur websocket qui fait la liaison entre les clients en temps réel.

Côté client, un script Javascript est exécuté dans le navigateur. Ce script permet de communiquer au serveur via websocket, et gère l'interface utilisateur propre à l'application. gourmand en terme de bande passante. Pour mettre en œuvre cette solution, plusieurs technologies sont nécessaires. Côté serveur, un serveur HTTP qui gère les connexions des utilisateurs, qui leur transmet les bonnes pages html ainsi qu'un serveur websocket qui fait la liaison entre les clients en temps réel. Côté client, un script javascript est exécuté dans le navigateur. Ce script permet de communiquer au serveur via websocket, et gère l'interface utilisateur propre à l'application.

### 3.2.2. Interface IHM

#### 3.2.2.1. Accueil

En arrivant sur le site, l'utilisateur verra la page d'accueil qui se présente sous une forme assez simple avec un champ de texte permettant de remplir son pseudonyme ainsi que 2 options: rejoindre un salon ou créer un salon. Pour rejoindre un salon l'utilisateur aura simplement à renseigner le code du salon qu'il souhaite rejoindre. Lors de la création d'un

salon, l'utilisateur aura juste à joindre le document sur lequel il souhaite travailler puis à partager le code au lecteur.

C.F [Annexe 1](#)

#### 3.2.2.2. Vue lecteur

Au niveau de l'interface pendant un cours, dans la fenêtre centrale sera affiché le pdf du cours, les lecteurs pourront se déplacer librement sur le document comme sur un pdf classique à la différence près que celui-ci peut être annoté par le présentateur. Dans le salon il y a deux rôles différents à savoir : les lecteurs qui représenteront les élèves et qui auront simplement le droit de regarder le cours sans pouvoir l'éditer et le présentateur (de base celui qui crée le salon) qui représente le prof qui a le droit d'annoter le pdf. Le système de synchronisation des pages sera basé sur la vue du présentateur donc il ne peut en avoir qu'un seul.

Ensuite nous allons retrouver au niveau de la barre de défilement de l'utilisateur un curseur rouge représentant la scrollbar du présentateur pour connaître l'avancée du cours.

Dans une fenêtre sur la droite de l'écran nous trouverons en bas à droite un bouton permettant de rejoindre rapidement la position du présentateur et une case à cocher pour activer la synchronisation de la position de la scrollbar avec celle du présentateur. Un peu au-dessus de ces deux boutons nous trouverons un champ texte accompagné d'un bouton "envoyer" correspondant au bouton question, la question devra être écrite dans ce champ texte et après l'envoi la question apparaîtra sur la partie haut droite de la fenêtre sous la forme d'une case de notification.

Il y aura une dernière fenêtre sur la gauche de l'écran avec la liste des noms des utilisateurs présents dans le salon. Ces deux fenêtres latérales pourront être réduites pour laisser plus de place à la fenêtre centrale.

C.F [Annexe 2](#)

#### 3.2.2.3. Vue présentateur

La vue présentateur sera assez similaire à la vue lecteur à la différence que les éléments de synchronisation du document pdf seront absents ainsi que le bouton "poser une question", et dans la partie haute de la fenêtre de gauche sera ajouté des éléments permettant de gérer le système de quizz.

C.F [Annexe 3](#)

### 3.3. Gestion du travail

La réalisation de ce projet demandant une quantité conséquente de travail, la mise en place de méthodes et d'outils est indispensable à l'avancement du projet. Voici quelques solutions mises en place pour organiser le développement du projet.

### 3.3.1. Discord

Discord est un logiciel de chat vocal et écrit très populaire avec notamment la possibilité de créer plusieurs salons. La grande majorité de nos communications passe par ce logiciel avec différents salons textuels qui nous permette d'avoir accès directement aux informations d'une catégorie en particulier.

### 3.3.2. Trello

Le modèle scrum n'est pas adapté pour notre projet, car nous avons très peu de créneau de travail en commun. Nous nous répartissons les tâches avec un délai et chacun s'organise comme il peut pour faire le travail. Cependant un outil d'organisation et de répartition du travail est nécessaire. Pour ce projet, nous utilisons l'outil en ligne Trello, car c'est un outil que nous avons utilisé notamment pour les projets des semestres précédant et avec lequel nous sommes familier.

Nous avons découpé notre trello en trois catégories : A faire, En cours, Fait. Cet outil nous permet de découper le travail en petites tâches à effectuer, puis d'y ajouter un membre et une date butoir (ainsi que pleins d'autres fonctionnalités que nous n'utilisons pas pour l'instant). Cette méthodologie permet à tout le monde de voir ce sur quoi les autres sont en train de travailler. Nous gardons les tâches effectuées car cela nous permet de voir a posteriori qui a effectué la tâche et d'aller voir cette personne en particulier en cas de problème ou de questions. De plus, ce logiciel est très visuel et permet de repérer très facilement, notamment avec un code couleur, quelle tâche nous concernent et à quelle catégorie elles appartiennent.

### 3.3.3. Tests

Pour un projet de cette envergure, le développement de tests en même temps que le code est indispensable. Le modèle TDD (test driven development), qui est basé sur le fait de développer les tests avant le code, n'est pas imposé dans le groupe car nous n'avons eu aucune formation à ce sujet, mais il est encouragé. Cependant, le développement de tests est obligatoire, ces tests permettent de vérifier le bon fonctionnement du projet, même après modifications. Ils aident également à identifier la source de bug, et peuvent même servir à comprendre le fonctionnement de code écrit par un autre membre du groupe, en voyant celui-ci en application.

### 3.3.4. Répartition des tâches

Pour la première partie, nous avons discuté tous ensemble du projet et avons réfléchi communément aux différents enjeux et contraintes du projet. Puis nous nous sommes réparties les tâches pour la rédaction du compte rendu.

## 3.4. Description d'un modèle de développement

Comme expliqué précédemment, l'utilisation d'une méthode Scrum n'est pas adaptée à notre projet car il n'est pas possible ni justifié d'organiser des réunions quotidiennes. Cependant nous allons essayer d'utiliser un maximum d'éléments des méthodes agiles. Notamment en fonctionnant par cycle et en travaillant sur un modèle



itératif. Nous aurons tout de même des réunions fréquentes (Plusieurs fois par semaine) pour que tout le monde puisse communiquer son état d'avancement.

Le projet sera découpé en plusieurs parties (lecteur pdf, serveur, ihm...) qui travaillent ensemble.

Le découpage par partie permet d'avoir des "spécialistes" d'un domaine qui connaissent très bien le code et la technologie sur laquelle il travaille.

# Annexe

## Annexe 1

Pseudonyme :

## Annexe 2

<p>Administrateur :</p> <p><input type="radio"/> Admin</p> <p>Invités :</p> <p><input type="radio"/> Invité1</p> <p><input type="radio"/> Invité2</p> <p><input type="radio"/> Invité3</p> <p><input type="radio"/> Invité4</p> <p><input type="radio"/> Invité5</p> <p><input type="radio"/> Invité6</p> <p><input type="radio"/> Invité7</p> <p><input type="radio"/> Invité8</p> <p><input type="radio"/> Invité9</p> <p><input type="radio"/> Invité10</p> <p><input type="radio"/> Invité11</p> <p><input type="radio"/> Invité12</p>	<p style="text-align: center;"><b>Titre</b></p> <p>Chapitre I</p> <p>*****</p> <p>*****</p> <p>*****</p> <p>*****</p> <p>*****</p> <p>*****</p> <p>*****</p> <p>*****</p> <p style="text-align: right;">1</p>	<p>Question 1</p> <p>Question 2</p> <div><div>Envoyer</div><div>Posez votre question</div></div> <p>Rejoindre Suivre <input type="checkbox"/></p>
--	--	--

## Annexe 3

Administrateur :

☐ Admin

Invités :

☐ Invité1  
☐ Invité2  
☐ Invité3  
☐ Invité4  
☐ Invité5  
☐ Invité6  
☐ Invité7  
☐ Invité8  
☐ Invité9  
☐ Invité10  
☐ Invité11  
☐ Invité12

Chapitre IV

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

3

Question 1

Question 2

Lancer un quizz

+