****

Quentin BADUEL

Fiona CASTELLI

Emmanuel YAGAPEN

Rapport de Synthèse

**MSOFT CINE**

Projet Industriel IG4

# Remerciements

Nous tenons à remercier l’ensemble des personnes qui nous ont permis de mener à bien ce projet pendant ces deux mois.

Nous remercions avant tout **M. CHAUSSE**, notre demandeur, pour nous avoir confié ce projet, ainsi que pour sa disponibilité et sa collaboration.

Nous tenons également à remercier **M. STRATULAT**, notre tuteur, pour ses conseils durant le projet.

De plus, nous remercions **M. DINU** pour son temps, son aide et ses conseils en ce qui concerne les technologies utilisées.

Finalement, nous tenons à remercier l’ensemble de l’équipe pédagogique, pour leurs enseignements, et du personnel administratif, pour leur accompagnement tout au long du projet.

Sommaire

[Remerciements 1](#_Toc294106027)

[Introduction 3](#_Toc294106028)

[I) Présentation du projet 4](#_Toc294106029)

[a. Le contexte 4](#_Toc294106030)

[b. Le demandeur 4](#_Toc294106031)

[c. Liste des utilisateurs 5](#_Toc294106032)

[d. La mission 6](#_Toc294106033)

[e. Contraintes et périmètre 6](#_Toc294106034)

[f. Objectifs détaillés 7](#_Toc294106035)

[II) Déroulement du projet 10](#_Toc294106036)

[a. Démarche 10](#_Toc294106037)

[b. Communication et synchronisation 10](#_Toc294106038)

[c. Ressources 11](#_Toc294106039)

[d. Estimation des charges 11](#_Toc294106040)

[e. Planification 12](#_Toc294106041)

[III) Résultats 14](#_Toc294106042)

[a. Travail effectué 14](#_Toc294106043)

[b. Limites 18](#_Toc294106044)

[c. Perspectives d’évolution 19](#_Toc294106045)

[IV) Difficultés rencontrées 20](#_Toc294106046)

[a. Techniques 20](#_Toc294106047)

[b. Ergonomie 22](#_Toc294106048)

[c. Gestion du temps 23](#_Toc294106049)

[V) Conclusion 24](#_Toc294106050)

[a. Bilan 24](#_Toc294106051)

[b. Discussion 24](#_Toc294106052)

[Webographie 25](#_Toc294106053)

[Documentation Python: 25](#_Toc294106054)

[Documentation Django: 25](#_Toc294106055)

[Documentation multi-langages 25](#_Toc294106056)

[Glossaire 26](#_Toc294106057)

# Introduction

Dans le cadre de notre formation de 4ème année en école d’ingénieurs à Polytech’ Montpellier, nous avons effectué un projet industriel proposé par M. Chausse, représentant de l’entreprise **IDEV**, société de prestation de services aux entreprises.

En groupe de trois étudiants nous avons mis en application, à l’aide d’un responsable pédagogique, nos acquis en gestion de projets ainsi que nos connaissances en informatique afin de répondre aux besoins du demandeur.

Le projet a débuté le 11 avril pour prendre fin le 27 mai, soit une durée de 6 semaines.

La mission confiée repose sur l’**élaboration d’une plateforme collaborative pour le domaine de l’audiovisuel et du cinéma**. En effet, la problématique actuelle repose sur les difficultés rencontrées lors des tournages telles que gérer les modifications régulières de scenario ou le fait de ne pas savoir où se trouvent le personnel de tournage ou le matériel nécessaire.

Ce rapport vise à expliquer de manière synthétique le travail accompli durant ces 6 semaines de projet. Pour plus de détails, nous vous invitons à vous reporter au rapport technique.

A travers cette synthèse, nous présentons dans un premier temps, le contexte général de notre projet industriel pour ensuite expliciter la mission confiée.

Puis dans une deuxième partie, nous traitons de la méthodologie employée pour mener à bien le projet.

La troisième partie expose le travail effectué, c’est-à-dire les différentes fonctionnalités développées, les difficultés rencontrées et les tests réalisés dans le but d’améliorer la performance de nos travaux.

Finalement, nous effectuons un bilan sur le projet puis nous ouvrons une discussion sur la place des systèmes d’informations dans le monde du cinéma.

# Présentation du projet

## Le contexte

Dans le monde du cinéma, il n’existe actuellement aucun logiciel permettant aux équipes de tournage de se synchroniser. Ainsi, un des problèmes majeurs rencontrés lors des productions vient du **temps perdu** par l’ensemble du personnel à attendre l’arrivée d’une personne ou d’un matériel.

On cherche donc à **résoudre des problèmes d’organisation** et de **synchronisation** en améliorant le système d’information existant.

Sur les tournages actuels, les informations sont échangées à travers des documents papiers ou de manière informelle. On souhaite ainsi mettre en place un outil informatique pour centraliser les données et optimiser les tâches.

Celui-ci se présentera sous la forme d’une **plateforme d’applications accessible sur Internet** et qui permettra aux différents utilisateurs d’accéder aux modules dont ils ont besoin.

Finalement l’utilisation de cet outil devra permettre de s’assurer que tout le monde dispose de toute l’information et que l’on puisse connaître les personnes qui n’ont pas été informées des dernières modifications afin de pouvoir les contacter.

## Le demandeur

M. Chausse est le représentant et le président de l’entreprise **IDEV** Sarl dont le siège social se trouve dans le Gard. IDEV est une société de prestation de services aux entreprises et a également une activité de « holding[[1]](#endnote-1) ».

Parallèlement, M. Chausse est aussi le président de **PROCINVEST**, société d’investissement, de la société **ARKAN SAS**, ayant une activité de « holding » et l’entreprise **CGMI SA**, qui a pour activité la fabrication de matières plastiques de base.

De par son activité, M. Chausse est en lien avec l’industrie du cinéma. Il recherche en effet des fonds pour la production de films.

## Liste des utilisateurs

Les principales personnes ayant propension à utiliser **l’application sur le terrain** (mobile) sont les techniciens, les accessoiristes, les acteurs ainsi que les responsables du film.

Il peut être intéressant pour ces personnes de connaître la géolocalisation d’autres individus participant au tournage.

Le **site web** sera surtout consulté par les réalisateurs, les scénaristes, les secrétaires ainsi que les services administratifs (par exemple: la comptabilité). Ils pourront non seulement s’informer la géolocalisation des autres personnes participant au tournage mais en plus définir et connaître les horaires de tournage. Les réalisateurs et scénaristes peuvent également utiliser l’application de gestion de scénario.

## La mission

Pour pouvoir à terme développer une **plateforme** efficace permettant la **gestion des informations relatives au tournage**, il est nécessaire de réaliser un **prototype** afin d’étudier son utilisabilité. En effet, les fonctionnalités à intégrer sont nombreuses et doivent être accessibles par différents utilisateurs.

La plateforme devra offrir à chaque utilisateur une interface dédiée, en fonction de son rôle au sein du tournage.

La plateforme contiendra 5 applications principales.

Notre mission consiste tout d’abord à réaliser la structure de cette plateforme. Une application permettant de gérer le scénario (suggestions et versions) est également inclue dans la mission.

Il est également question de concevoir une seconde application pour gérer les horaires de présence et la géolocalisation des membres d’une équipe de tournage. Cependant, cette partie de la mission dépend du temps restant après la réalisation de la première application.

Les autres applications qui pourront être mises en place à la suite de notre projet concernent le suivi des tournages, le suivi des budgets et le suivi des conditions de tournage (météo, équipement,…)

Les fonctionnalités devront être **accessibles depuis différents supports** (PC, tablettes mobile et Smartphones) et sur différents navigateurs.

## Contraintes et périmètre

L'élaboration d'une plateforme Web requiert l'utilisation de nombreuses **technologies Web** dont la plupart nous sont inconnues.

Afin de permettre une réalisation rapide, il a été décidé d’après les conseils de M. Dinu, d'utiliser le framework **Django**[[2]](#endnote-2) et un service d’hébergement de site web fourni par **Google App Engine**[[3]](#endnote-3).

Le développement doit donc se faire en utilisant les langages utilisés par Django (notamment Python[[4]](#endnote-4)). La partie interface Web intègre également les technologies HTML[[5]](#endnote-5), CSS[[6]](#endnote-6) et JavaScript[[7]](#endnote-7).

Au départ, le périmètre de la mission était très vague et prévoyait cinq applications. Puisqu’il n'était pas possible d'achever en si peu de temps un prototype aussi complet, nous avons décidé d’un commun accord avec le demandeur, de se concentrer sur deux applications principales.

## Objectifs détaillés

### Objectifs de l’application 1

L’application devra permettre aux scénaristes et aux réalisateurs de **se mettre d’accord sur une version du scénario** qui sera par la suite communiquée à l’ensemble des équipes.

On définit une **suggestion** comme étantune **proposition** de scénario envoyée par le scénariste et qui n’a pas été validée. Une **version** désigne quant à elle une proposition qui a été validée.

Actuellement, les scénaristes envoient des propositions de scénario à un réalisateur qui lui renvoie éventuellement des remarques et des demandes de modifications. Lorsque les scénaristes et le réalisateur se sont mis d’accord sur une version, celle-ci est imprimée en un grand nombre d’exemplaires afin d’être distribuée.

Le scénario peut être **révisé plusieurs fois par jour**, mais il doit être réimprimé intégralement pour que chaque membre de l’équipe dispose toujours de la dernière version et donc de la même numérotation de pages.

Il faudra donc gérer de manière efficace les **informations échangées** entre les réalisateurs et les scénaristes. Il sera également nécessaire de garder toutes les traces de ces échanges pour établir un historique.

De plus, afin d’éviter des impressions conséquentes et systématiques de pages, il faudra permettre aux membres des équipes de **télécharger**, depuis la plateforme, la version courante du scénario.

Un système de **notifications** et de **relances** devra également être mis en place afin de s’assurer de la bonne réception des nouveaux scénarios.

L’application devra intégrer quelques contraintes. En effet, il faudra pouvoir décomposer les propositions de scénario en scènes pour pouvoir suivre l’historique des versions et **associer à chaque scène les informations nécessaires au tournage** (les acteurs, le matériel...).

Le **flux opérationnel** (« *workflow* ») de l’application 1 peut se définir tel que suit :

1. Le réalisateur poste la première version du scénario. Il peut alors choisir d’émettre une correction afin que le scénariste en soit notifié.
2. Le scénariste émet une suggestion. Le réalisateur est alors notifié.
3. Le réalisateur peut choisir de proposer une correction ou de valider la proposition. S’il la valide, la proposition devient une version et l’équipe de tournage est notifiée.
4. Lorsque les membres de l’équipe de tournage téléchargent la version du scénario ; ils accusent réception automatiquement.

L’assistant de production peut relancer les individus n’ayant pas accusé réception via une interface dédiée

*Ci-dessous, un schéma récapitulatif de l’utilisation de l’application 1 :*

8 : Consultation

des Accusés

Réception



Scénariste

Réalisateur

**1 :** Envoi de

suggestion

Staff

**2** : Mise en

ligne

**4** : Validation de la

suggestion

**9** : Relance

**6** : Notification

**7** : Téléchargement

5 : Création du PDF

**3** : Notification

Assistant de production

**PLATEFORME**

### Objectifs de l’application 2

Cette application peut être décomposée en deux fonctionnalités distinctes.

La première consistera à **valider la présence des personnes travaillant sur le tournage** afin de comptabiliser leurs heures de travail (qu’elles soient présentes ou non sur le lieu du tournage). En effet les horaires et les conditions de travail varient beaucoup lors d’un tournage. Par exemple, un membre de l’équipe peut travailler de nuit sur des sites différents ou même de chez lui.

Cette fonction permettra de connaître les horaires de travail effectués par un utilisateur et d’aider le service de la comptabilité à établir les bulletins de paie.

La deuxième fonctionnalité se basera sur les signaux GPS intégrés dans les téléphones des utilisateurs et dans certaines balises attachées au matériel.

Les responsables du tournage pourront alors **connaitre à tout moment et à l’aide d’une carte la position d’un matériel ou d’un membre de l’équipe de tournage** dès que le système de traçage sera activé.

*Voici un exemple du fonctionnement de l’application 2 :*



**PLATEFORME**



**1** : Recherche de la géolocalisation

d’un membre du tournage

Réalisateur

**2** : Envoi d’un signal

au Smartphone

**3** : Envoi des coordonnées GPS

**4** : Point sur la carte indiquant

la géolocalisation de la personne

# Déroulement du projet

## Démarche

Dès le début du projet, nous avons rapidement cerné les besoins du demandeur et nous avons par la suite effectué une **phase d’analyse** d’environ une semaine durant laquelle nous avons établi un cahier des charges et élaboré la conception générale.

Parallèlement, nous avons étudié les outils dont nous avions besoin afin d’avoir une idée plus précise de la mission et ainsi débuter notre planification.

Afin de s’assurer que notre compréhension de la mission était correcte nous avons réalisé deux **réunions avec le demandeur** et nous avons correspondu par e-mail afin de préciser certains points.

Dans le même temps, nous avons rédigé plusieurs documents pour formaliser les besoins exprimés (lettre de mission, diagramme de cas d’utilisation, d’activité, de séquence système). A partir de ces besoins nous avons pu modéliser les fonctionnalités à élaborer à travers des diagrammes fonctionnels (diagrammes de séquence).

On a ainsi débuté la **phase de développement et tests** à la fin de la deuxième semaine.

Afin d’optimiser notre temps durant cette phase, nous nous sommes répartis le travail par fonctionnalités et par interfaces.

## Communication et synchronisation

Afin de pouvoir partager rapidement le résultat de nos travaux, nous avons décidé de **réaliser ce projet de la façon la plus collaborative possible.**

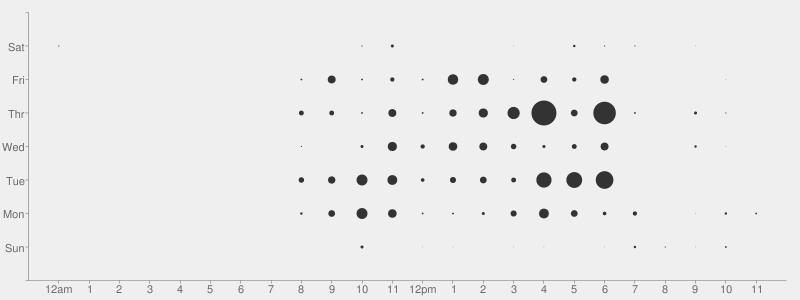
Nous avons ainsi utilisé les outils Google Documents[[8]](#endnote-8) pour le partage de la documentation, Github[[9]](#endnote-9) pour le partage du code et de documents, et Remember The Milk[[10]](#endnote-10) pour le partage des tâches.

De plus la possibilité de disposer d’une salle de projet a été une excellente occasion pour travailler ensemble et communiquer directement pendant ces six semaines.

Concernant notre propre communication et synchronisation au sein du groupe, elle a pu se faire facilement puisque nous étions tous trois présents aux mêmes horaires dans une même salle. De plus, nous nous entendons bien.

Ainsi, nous ne nous sommes pas attribués de rôles fixes mais avons alterné. Cela dans deux buts : manipuler l’ensemble des technologies et pouvoir prendre du recul.

Ci-dessous, un graphe présentant les flux de mise en ligne des codes sources sur le serveur.



Graphe des « commit »sur Github par jour et heure

## Ressources

Pour mener à bien ce projet, nous avons constitué une équipe de trois élèves-ingénieur avec un **planning de 28 jours**. Nous avons été soutenus par deux tuteurs, **Tiberiu Stratulat** et **Razvan Dinu**.

Nous avons pu disposer d'une salle de projet fournie par Polytech’ Montpellier et équipée de deux ordinateurs et quatre écrans. Nous avons également utilisé nos ordinateurs personnels.

M. Chausse, notre demandeur, nous a gracieusement prêté trois tablettes tactiles fonctionnant sous Android.

## Estimation des charges

Nous nous sommes rapidement aperçu que l’ensemble des fonctionnalités demandées étaient bien trop important pour que nous puissions les réaliser dans le temps imparti. Après discussion avec le demandeur, nous avons donc restreint les objectifs à deux applications.

Nous avons ensuite **estimé le** **temps nécessaire pour l’apprentissage des nouveaux outils**, des **phases de conceptions** et de **développement**. Ces estimations ont été assez “imprécises” car elles reposent que sur nos expériences préalables, mais elles nous ont permis d’organiser de façon générale le projet.

## Planification

La délimitation des tâches nous a permis d’établir une planification que nous avons formalisée grâce à Microsoft Project[[11]](#endnote-11).

Nous avons commencé par définir le nombre exact de jours dont nous disposions en ignorant les demi-journées de cours et les week-ends.

Cela nous a permis de procéder à une estimation de chaque partie du projet. Nous les avons ensuite décomposés en tâches pour obtenir une estimation du temps nécessaire pour les différentes charges.

Il a également fallu répartir les jours-homme de manière équitable entre les différents membres du groupe.

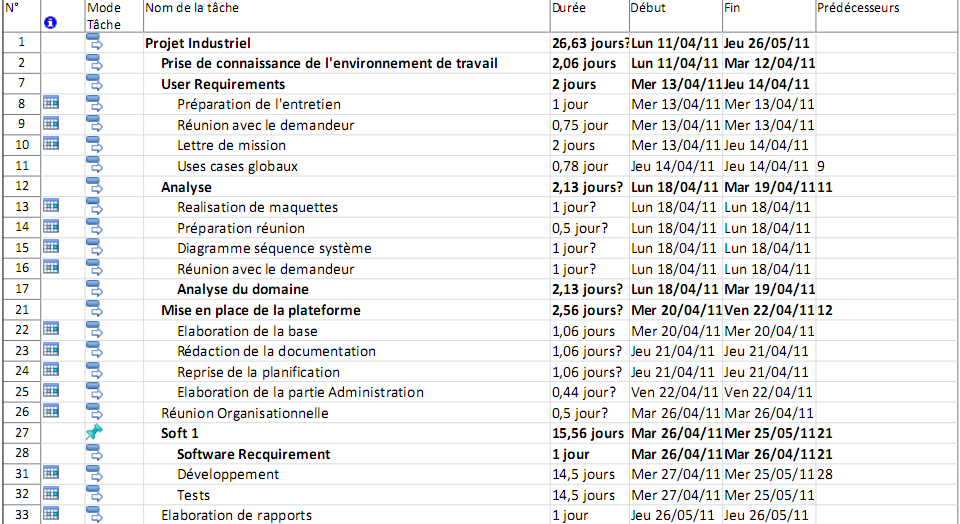
Nous avons présenté notre **première version de la planification** à M. Ruiz (responsable du cours sur la gestion de projets) le 26 avril 2011, ce qui nous a permis de reprendre notre planification pour l'améliorer. En effet nous avons dans un premier temps essayé de planifier les taches en fonction du temps imparti, au lieu de calculer le temps nécessaire pour effectuer toutes les tâches.

Dans un deuxième temps, en milieu de projet nous avons **réajusté la planification** pour nous rendre compte que nous ne pourrions pas remplir tous les objectifs initiaux dans le temps imparti.

Nous avions prévu cette éventualité dès le début du projet, et nous avons gardé le contact avec notre tuteur et notre demandeur qui ont compris nos difficultés. Cette re-planification a permis de mettre au clair notre avancement sur le projet et de reconsidérer le temps nécessaire pour les tâches restantes.

Nous avons décidé d'abandonner certaines fonctionnalités de la plateforme pour nous **concentrer sur l'application 1 et la rendre plus aboutie que la version prévue initialement**.

Ainsi, nous avons par exemple mis en place le tableau de bord permettant de récupérer l’ensemble des informations associées au scénario et au tournage.



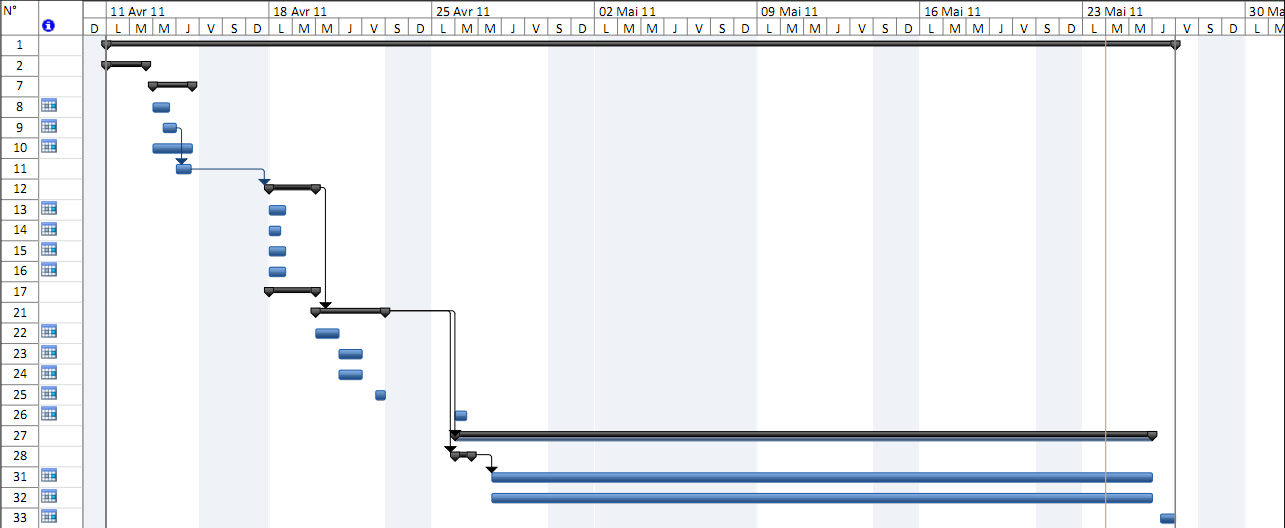


Diagramme de Gantt du projet

# Résultats

## Travail effectué

### Conception

Afin de procéder à la phase de conception, nous avons utilisé le langage de modélisation **UML** spécifique aux méthodes orientées objet.

Pour ce projet, nous avons réalisé un dossier de conception. Celui-ci contient un **diagramme de classe**, l'ensemble **des diagrammes de cas d'utilisation**, les **diagrammes d'activité**.

Le diagramme de classe représente l’ensemble des entités de la base de données de la plateforme. Il sera utilisé pour bâtir le modèle.

Le diagramme de cas d’utilisation est une description du modèle « vue » par les acteurs du système. Elle correspond aux besoins attendus par chaque utilisateur. Ce diagramme est très utile pour cerner les besoins et souhaits du demandeur.

Le diagramme d’activité permet de décrire sous la forme d'enchaînement d'actions le comportement du système.

Ces diagrammes résument de façon **synthétique** le besoin exprimé par l’utilisateur. De plus, cela permet d’en discuter directement avec le demandeur qui n’est pas forcément familier avec l’aspect technique.

Nous avons de plus réalisé les **spécifications fonctionnelles** pour l'application 1 (diagrammes de séquence, diagramme d’état).

Le diagramme de séquence est une représentation séquentielle des interactions entre les éléments du système et/ou de ses acteurs.

Le diagramme d’état permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.

Pour plus d'informations, vous pouvez vous référer au rapport technique.

### Développement

#### Design

Nous avons accordé une importance particulière à **l’interface graphique de la plateforme**. En effet, nous avons privilégié des Interfaces Homme-Machine (IHM) légères et intuitives. En effet, étant donné que la plateforme sera consultée à partir de tablettes tactiles ou de Smartphones, il faut que les IHM soient synthétiques et fonctionnelles.

Cependant, n’ayant pas beaucoup d’expérience dans ce domaine, nous avons dû demander conseil à notre tuteur, M. Dinu. Il nous a alors conseillé l’utilisation d’outils qui nous ont aidés à améliorer les interfaces de façon très simple.

#### Plateforme

La plateforme comprend l’**administration des objets** présents dans la base de données et la gestion des profils des utilisateurs. C’est une base où on pourra intégrer plusieurs applications. Chaque utilisateur aura accès à des interfaces personnalisées, en fonction de son profil.

L’élaboration de cette plateforme a pris 3 jours et la plus grande difficulté fut de créer un modèle capable d’accueillir les différents objets décrits dans le diagramme de classe.

La partie administration est **une interface de contrôle** où on peut créer, consulter ou modifier les objets du modèle. Nous avons utilisé les fonctions fournies par le Framework Django pour mettre en place l’interface d’administration. Afin de rendre les IHM plus fonctionnelles il faut redéfinir certains affichages et lier certains objets entre eux. Par exemple, à travers l’interface d’administration d’une proposition, on a accès toutes les remarques qui y sont associées.

On a également modifié les objets « Users » prédéfinis dans le Framework Django afin de pouvoir leur associer des informations comme le scénario sur lequel il travaille.

#### Application 1 : Scénario

L’application 1 permet une **gestion des propositions**, de leur **validation** et une **diffusion** des scénarios modifiés aux personnes concernés.

C’est l’application la plus cruciale de la plateforme car elle s’articule autour de l’élément le plus important d’une production audiovisuelle, le scénario. Ainsi, une attention particulière doit être apportée sur le développement de cette application, car il s’agit du **cœur de la plateforme**.

En effet, une fois la gestion des scénarios implémentée, on pourra **intégrer d’autres applications** autour de celle-ci, comme par exemple, la gestion des équipes de tournage propre à chaque scène.

Nous avons débuté le développement en mettant en place **l’interface scénariste** c’est-à-dire, la **mise en ligne des propositions** de scénario au format Microsoft Word 2007 (.docx).

Cette première fonctionnalité comprend également l’**extraction des scènes** et la **mise en évidence des différences** avec les précédentes propositions. De plus, nous avons également mis en place un mécanisme pour **l’hébergement de fichiers** afin de garder toutes traces des suggestions effectuées.

Cependant, comme nous utilisons des technologies nouvelles, le développement de ces différentes fonctions nous a pris beaucoup de temps, notamment pour la lecture de document Word et l’extraction des scènes du scénario.

En s’appuyant sur la mise en ligne des propositions, nous avons élaboré **l’interface réalisateur** avec la **mise en place de la première version du scénario** et la **validation des propositions**. La plus grande difficulté de cette IHM fut de gérer la validation des suggestions avec la notification de certains utilisateurs par mail.

Une fois la gestion des propositions et de leur validation implémentée, nous avons élaboré **l’interface secrétaire**. On a alors mis en place la **consultation des validations** et les personnes n’ayant pas **accusé réception** de celles-ci.

De plus, un **système de relance** a également été réalisé pour pouvoir notifier rapidement les utilisateurs qui n’ont confirmé la réception du nouveau scénario. Le développement de cette partie de l’application fut rapide, puisque toutes les fonctionnalités conçues étaient basé sur des fonctions existantes.

Finalement, nous avons implémenté **l’interface propre aux membres des équipes de tournage**, c’est-à-dire la **consultation et le téléchargement des différentes versions validées**.

Il a ainsi fallu mettre en place la génération de fichier PDF à partir de documents Microsoft Word. Pour cela, nous avons utilisé un convertisseur hébergé sur un site web. Nous utilisons des requêtes HTTP[[12]](#endnote-12) pour communiquer avec ce dernier.

#### L’historique des versions

Afin de pouvoir facilement visualiser les différentes versions d’un scénario, nous avons conçu un « **Dashboard[[13]](#endnote-13)**». Grâce à cette interface, nous pouvons consulter les différents scénarios retenus et les différences entre eux. Le scénario est découpé en scènes auxquelles on peut rattacher des informations comme le personnel concerné ou le matériel nécessaire pour le tournage.

De plus, des données générales sur le scénario sont présentées, comme la date de début et de fin du tournage ou encore le synopsis. Le développement de cette fonctionnalité a été réalisé en dernier et elle nous a permis de savoir si l’application 1 fonctionnait bien. En effet, tout dysfonctionnement dans la partie gestion des propositions et de leur validation aurait conduit à des incohérences dans la visualisation des éléments du « Dashboard ».

### Tests

Lors de la phase de développement, nous avons commencé par élaborer une interface d’administration permettant la gestion des utilisateurs et des différents objets du programme.

Nous avons ensuite créé un petit échantillon pour chaque objet de la base de données afin de pouvoir disposer de plusieurs éléments lors des tests que nous avons réalisé en même temps que la conception de l’architecture de la plateforme.

Nous avons décidé d’effectuer la majorité des tests au moment du développement. En effet notre travail portait essentiellement sur la conception d’une plateforme web, ainsi toute modification a eu un impact direct sur l’affichage.

Nous avons procédé en **développant les différentes fonctionnalités de façon hiérarchique**, c’est à dire que nous avons élaboré les bases avant de nous en servir. Cela nous a alors placés dans le rôle d’utilisateur des parties précédemment réalisées. Nous avons donc pu contrôler tout au long du développement que le code réalisé précédemment était fonctionnel.

Enfin, l’un de nos principaux objectifs était d’élaborer une **interface ergonomique et intuitive**, pour cela nous avons endossé le rôle d’un utilisateur afin de déterminer quels étaient les différents scénarios d’utilisation et nous les avons testés pour **contrôler qu’aucun bug important ne subsiste**.

### Documentation

#### Guide d’utilisation

Comme souhaité par le demandeur, nous avons livré un guide d’utilisation rédigé en anglais (« *user guide* »).

Celui-ci contient les parties suivantes :

* Mise en place de la plateforme
* L’interface d’administration
* L’interface réalisateur
* L’interface scénariste
* L’interface assistant de production
* L’interface « staff » (tous les autres profils).

Pour lire le contenu de ce guide d’utilisation, vous pouvez vous reporter aux annexes du rapport technique.

## Limites

Nous avons été confrontés à plusieurs limites parmi lesquelles nous pouvons citer les suivantes :

Google App Engine obligeant à travailler avec un seul langage de programmation, en l’occurrence nous avons utilisé Python. A cause de cette limite, certaines fonctionnalités sont difficilement implémentables. Une solution serait donc que la plateforme fonctionne sur un serveur avec Système d’Exploitation.

Pour exemple, du fait de ceci, nous nous sommes limités au traitement de fichiers .docx, alors que d’autres **formats de texte** seraient envisageables : .doc, .odt etc.

Concernant les fichiers de scénario, on peut les découper grâce à des balises ##SCENE [numéro de scène]##. Cependant, ce système est contraignant car il oblige à numéroter à la main chaque scène. On peut donc imaginer qu’il faudrait mettre en place un système d’auto-génération de la numérotation. Cette fonctionnalité pourrait par ailleurs prendre en charge la numérotation lors de l’insertion de nouvelles scènes.

Une autre limite est la **gestion de l’informel**. En effet, si par exemple le réalisateur communique ses commentaires et remarques de manière informelle, le scénariste est bloqué car l’envoi de nouvelles suggestions est dépendant de l’envoi d’une correction ou d’une validation de la part du réalisateur.

Egalement, du fait que le site soit toujours à l’état de prototype, pour **initialiser le système** actuellement il faut passer par l’interface d’administration pour définir certains paramètres. Par exemple, on définira le scénario et les rôles des utilisateurs.

Enfin, hormis auprès du demandeur M. Chausse, nous n’avons pas pu faire tester l’application par les personnes susceptibles d’utiliser la plateforme plus tard.

## Perspectives d’évolution

Il existe de nombreuses perspectives d’évolution concernant la plateforme MSoft Cine.

Tout d’abord, il faudra **ajouter d’autres applications et fonctionnalités** utiles durant le tournage. On peut citer notamment :

* Le suivi de présence du personnel
* La géolocalisation du personnel et du matériel
* Le suivi des tournages quotidiens (avec visionnage vidéo)
* Le suivi des budgets
* Un tableau de bord informant de données pertinentes (conditions météo, équipements disponibles)

Enfin, la mise en place d’une version **mobile** du site et/ou une application mobile sont à envisager, étant donné qu’un usage régulier de la plateforme sera fait depuis des terminaux mobiles (tablettes et Smartphones)

Détaillons à présent **les évolutions possibles pour l’application 1** (gestion du scénario).

On peut vouloir commenter et faire des remarques dans les fichiers .docx à l’aide de l’outil déjà inclus dans Microsoft Word (outil « Révision »). Il faudrait alors pouvoir les prendre en compte sur l’application (analyse du fichier « *comment.xml* »).

Concernant les scènes, il serait souhaitable de pouvoir les modifier directement sur la plateforme avec un outil d’édition en ligne. Pour cela, on peut par exemple utiliser [l’application Django](http://pypi.python.org/pypi/django-tinymce/) fournissant l’interface [TinyMCE](http://tinymce.moxiecode.com/).

Concernant le téléchargement de scénario, on peut imaginer vouloir télécharger les scènes individuellement ou uniquement certaines scènes choisies.

# Difficultés rencontrées

## Techniques

### Apprentissage des technologies

Pendant ce projet industriel, nous avons dû utiliser simultanément plusieurs outils, ce qui nous a permis de disposer d’un **grand nombre de** fonctionnalités afin de réaliser nos objectifs.

En effet, Django nous a fourni une base de travail facilement configurable pour créer l’interface d’administration. JQuery a permis de simplifier les commandes JavaScript et de rendre les interfaces plus jolies. Github quant à lui nous a permis de tous participer au développement simultanément.

Cependant cela, a induit que nous devions assimiler rapidement le fonctionnement de ces différents outils.

Par exemple, concernant le Framework Django, bien que la structure de base soit facile à appréhender, l'utilisation des fonctions courantes peut rapidement devenir difficile. En effet **Django présente trois syntaxes différentes** en fonctions des modules utilisés (interfaces web, les requêtes de la base de données et les traitements).

Malgré ceci, nous avons su nous **adapter** et surtout, **voir l’intérêt** des différentes technologies utilisées.

Par exemple Google App Engine a pour intérêt d’héberger notre application très facilement et d’y apporter des modifications. De plus son interface d’administration est également bien conçue (consultation des logs, accès à la base de données,…)

Django quant à lui, permet de réduire la quantité de code nécessaire pour réaliser des pages web. En effet grâce à son système de template, il est possible d’utiliser la notion d’héritage. Par exemple, on peut réutiliser la structure, les éléments et le style d’une page HTML pour créer d’autres pages en redéfinissant qu’une partie du contenu. De plus Django permet de séparer le modèle des traitements et de l’interface, ce qui garantit une architecture solide et efficace.

### Problème de compatibilité entre les applications

Nous avons été confrontés à plusieurs problèmes au niveau de la compatibilité entre les différents outils utilisés.

#### Google App Engine

Lors de la phase d'installation de la base de la plateforme, nous avons voulu utiliser le format proposé par Django qui permet d'intégrer dans le modèle de données des relations « ManyToMany »[[14]](#endnote-14) entre les éléments.

Cependant bien que cette solution marche lors de la phase de développement en local, nous nous sommes aperçus que ce **format de base de données n'est pas supporté par Google App Engine**.

Nous avons donc repris notre modèle de conception en remplaçant toutes les relations de « un à plusieurs » ou de « plusieurs à plusieurs » par des **classes intermédiaires** contenant la concaténation des références vers les objets concernés (solution pour un schéma relationnel classique).

De plus, lors de la phase de développement où nous avons voulus effectuer des **opérations entre les objets**, nous avons constaté que Google App Engine ne peut pas établir des jointures entre certains éléments de la base de données (GAE ne peut pas effectuer sur des objets identifiés par des clés étrangères). Pour résoudre ce manque, nous récupérons tous les objets d'un type et effectuons un parcours sur chaque instance pour effectuer des tests de comparaisons.

De plus, Google App Engine offre un **environnement sans système d’exploitation** pour héberger des applications en Python. Cependant cette solution ne permet pas de faire appel aux applications dans un autre langage que Python permettant d’effectuer des taches telles que le l’analyse et le découpage (*parsing*) de fichier doc, ou la transformation d’un fichier .docx en .pdf.

De ce fait nous avons dû **rechercher ou concevoi**r nous-même des **solutions uniquement écrites en Python** et qui soient indépendantes du système d’exploitation ou passer par des **services externes** accessibles à partir de requêtes HTTP.

Mais, cette implémentation permet d’assurer une certaine **dépendance** de l’application par rapport au système d’exploitation et de la configuration du serveur, ce qui confère au programme une plus grande **portabilité**.

#### Problème sur les formats

Pendant l’élaboration de l’application 1, nous avons connu des problèmes concernant la compatibilité des formats. En particulier pour la gestion des dates.

En effet, les comparaisons de dates peuvent poser problème car un même objet peut se présenter d'un grand nombre de façons différentes.

Pour effectuer les comparaisons avec les dates entrées par l'utilisateur il est nécessaire d'effectuer des conversions. En effet, par exemple le format de **Django enregistre en années celles après l'an 2000**. Ainsi, dans le format Django nous sommes en l'an 11.

Le problème des dates est devenu encore plus sensible lorsque nous avons intégré du JavaScript pour faciliter la sélection des dates. En effet en JavaScript, **les dates commencent à partir de 1900** et janvier correspond au mois 0 et non pas au mois 1.

On s’est donc assuré de vérifier à chaque actualisation si une date a été choisie ou si elle doit être auto-générée. Dans les deux cas, on doit modifier le format pour pouvoir effectuer des comparaisons avec les dates présentes dans la base de données et retrouver les suggestions concernées.

Nous avons aussi rencontré des problèmes avec les **formats de fichiers**. En effet, nous nous sommes d’abord orientés vers la manipulation de fichiers .doc. Cependant, ceux-ci étant codés de manière binaire il est apparu difficile de les parcourir et analyser (« *parsing* »).

Nous nous sommes donc concentrés sur le format .docx. Ce type de document est en fait une archive de fichiers xml. Nous avons donc pu « parser » le fichier en tenant compte des balises quand cela était nécessaire.

## Ergonomie

Nous avons consacré un temps important pour améliorer l'ergonomie du site en mettant en place des onglets pour **faciliter la navigation**. Nous avons intégré un système **d'affichage dynamique** pour la présentation des suggestions. La sélection des dates est quant à elle facilitée par un calendrier.

De plus **l'affichage est spécifique aux rôles**. Seules les scénaristes et les réalisateurs ont accès à toutes les suggestions, les utilisateurs normaux ne voient que les versions validées et les secrétaires peuvent visualiser la liste des personnes concernées et celles qui ont téléchargé ou non la dernière version.

## Gestion du temps

Lors de la planification, nous avions déjà prévu un temps important pour **l’apprentissage des technologies** liées au projet. Cependant celui-ci ne s’est pas avéré suffisant.

De ce fait, nous avons dû reprendre notre planification afin de ré-estimer certaines charges et de s’assurer qu’il nous serait possible d’assurer un livrable dans les temps impartis pour le projet.

# Conclusion

## Bilan

Ce projet de deux mois a été **riche en expériences** sur plusieurs niveaux. Tout d’abord, en tant que projet industriel puisque c’était notre premier projet professionnel en groupe. La relation école – entreprise est vraiment essentielle dans une formation d’ingénieur et cela nous a permis d’appréhender au mieux tous les aléas d’un tel projet.

De plus, **l’étude d’un système d’information** et son informatisation ont été très profitables étant donné qu’ils sont un point clef de notre formation.

Nous avons également pu tester notre aptitude à être responsables de projet, c’est-à-dire estimer les charges des différentes phases, gérer nos plannings et prendre des décisions. C’est réellement la meilleure façon de comprendre un tel métier et d’apprendre de nos erreurs.

Par la suite, ce projet nous a également permis de développer notre aptitude à travailler en équipe. Nous avons donc pu **mettre en application** nos capacités d’écoute, d’ouverture d’esprit, de sens relationnel ainsi que d’organisation et d’analyse.

Finalement, que ce soit sur nos réussites ou nos difficultés, ce projet industriel reste une étape enrichissante dans cette 2ème année de cycle d’ingénieur.

## Discussion

La place des Systèmes d’Informations dans le cinéma

Lors de ce projet, nous avons pu en savoir plus sur le monde du cinéma actuel.

Bien que cette industrie génère des investissements et des profits de dizaines voire centaines de millions d’euros, le système d’informations inhérent aux tournages est resté **très informel** et **sans réelle organisation**.

En effet, il semble que beaucoup de temps soit perdu à trouver les acteurs des scènes réalisées sur les plateaux de tournage (problèmes d’organisation). De même, la gestion de la relation scénariste-réalisateur concernant la modification du scénario est informelle. Cela entraine une réimpression systématique du scénario pour tout le personnel du tournage.

La multiplication des versions en format papier peut également poser un problème de synchronisation dans l’équipe.

Alors que le matériel de tournage est bien souvent à la pointe de la technologie, il semble important de **proposer une solution informatique** pour l’aide à la gestion de tournages.

# Webographie

## Documentation Python:

* « **A Quick, Painless Tutorial on the Python Language** » N. Matloff , 2010

<http://heather.cs.ucdavis.edu/~matloff/Python/PythonIntro.pdf>

## Documentation Django:

* **Site officiel de Django**

<http://docs.djangoproject.com/en/dev/intro/>

## Documentation multi-langages

* **Le site du Zéro**

<http://www.siteduzero.com/>

* **Développez.com**

<http://general.developpez.com/cours/>

* **Wikipedia**

<http://www.wikipedia.org/>

# Glossaire

1. **Holding :**

   Un ou une holding est une société ayant pour vocation de regrouper des participations dans diverses sociétés et d'en assurer l'unité de direction. [↑](#endnote-ref-1)
2. **Django** :

   Django est un frameworkxv de développement web en Python. Il repose sur l’architecture MVCxvi.

   Le but premier de Django est de faciliter la mise en œuvre de sites web complexes interfacés à des bases de données. Django peut être considéré comme une boite à outils où chaque module peut fonctionner de façon indépendante. [↑](#endnote-ref-2)
3. **Google App Engine** :

   Plateforme de [conception](http://fr.wikipedia.org/wiki/Programmation_Web) et d'hébergement d'[applications web](http://fr.wikipedia.org/wiki/Application_web), basée sur les [serveurs](http://fr.wikipedia.org/wiki/Serveur_informatique) de [Google](http://fr.wikipedia.org/wiki/Google). [↑](#endnote-ref-3)
4. **Python** :

   Python est un langage de programmation. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser. [↑](#endnote-ref-4)
5. **HTML** :

   Format de données conçu pour représenter les pages web, basé sur le principe de balises. [↑](#endnote-ref-5)
6. **CSS** :

   CSS (Cascading Style Sheets : feuilles de style en cascade) est un langage informatique qui sert à décrire la présentation des documents HTML. [↑](#endnote-ref-6)
7. **JavaScript** :

   JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives. [↑](#endnote-ref-7)
8. **Google Documents** :

   Google Documents est un ensemble de programmes permettant un travail en ligne collaboratif. [↑](#endnote-ref-8)
9. **Github** :

   Github est un service web d'hébergement et de gestion de développement de logiciels, utilisant le programme Git. Il sert avant tout comme outil de gestion de version et de travail collaboratif (semblable SVN). [↑](#endnote-ref-9)
10. **Remember the Milk** :

    Remember the Milk est un outil de gestion de tâches qui peut être collaboratif. [↑](#endnote-ref-10)
11. **Microsoft Project** :

    Microsoft Project (ou MS Project ou MSP) est un logiciel de gestion de projets édité par Microsoft. Il permet aux chefs de projet et aux planificateurs de planifier et piloter les projets, de gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets. [↑](#endnote-ref-11)
12. **Requête HTTP** :

    L'HyperText Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement le « protocole de transfert hypertexte », est un protocole de communication client-serveur.

    Les requêtes HTTP permettent au navigateur de demander et d’afficher des pages web. [↑](#endnote-ref-12)
13. **Dashboard :**

    Un dashboard peut être défini comme un tableau de bord présentant une vue synoptique du système. [↑](#endnote-ref-13)
14. **Many To Many** :

    En base de données, une relation « Many To Many » (plusieurs à plusieurs) est une realtion qui concerne deux objets A et B. A est lié à plusieurs objets B et B est lié à plusieurs objets A.

    xv**Framework**:

    En programmation informatique, un framework est un kit de composants logiciels structurels, qui servent à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel (architecture).

    Les frameworks sont utilisés pour modeler l'architecture des logiciels applicatifs, des applications web et des composants logiciels.

    xvi**MVC**:

    Le Modèle-Vue-Contrôleur (en abrégé MVC, de l'anglais Model-View-Controller) est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle. [↑](#endnote-ref-14)