

Conception et développement d'une plateforme d'hébergement d'événements

[Dossier d'analyse et de conception]

Équipe projet B1-16

- DECHENAUD Émile
- SEKMA Sarah
- HAYOT Lynn
- DUCRUET Léo
- TIVOLLIER Jarod
- DUBIEF Émilie

Table des matières

Consignes et évaluation du dossier de conception.....	2
Introduction.....	3
Expression des besoins et Analyse.....	4
Description des cas d'utilisation.....	4
Scénarios nominaux.....	6
Scénarios alternatifs.....	8
Priorisation des cas d'utilisation.....	10
Architecture proposée.....	11
Exemples d'utilisation.....	12
Conception détaillée.....	14
Architecture détaillée.....	14
Détail des scénarios.....	16
Conclusion.....	17

Consignes et évaluation du dossier de conception

Ce dossier de conception est un template à partir duquel vous pouvez vous inspirer afin de réaliser votre dossier d'analyse et de conception. Les indications entre crochets sont là pour vous aider à structurer votre dossier. Elles sont à retirer dans le rendu final. Seront évalués :

- La pertinence et la validité des modèles créés ;
- La conformité des modèles entre eux (par exemple le diagramme de séquence détaillé par rapport aux scénarios nominaux) ;
- La conformité du code par rapport aux modèles créés ;
- Le respect de la notation UML ;
- La clarté et la qualité du dossier de conception, y compris de l'introduction, de la conclusion, et de la description que vous ferez des différents modèles.

Quelques conseils afin d'améliorer la clarté de votre dossier :

- Ne pas se contenter d'insérer des images des modèles créés, mais les accompagner d'explications textuelles (même brèves) ;
- Apporter une attention particulière à la clarté des diagrammes réalisés :
 - ☒ Vérifier la résolution et la lisibilité des diagrammes ;
 - ☒ Vérifier que la notation UML est parfaitement respectée ;
 - ☒ Ne pas hésiter à éclater un diagramme en deux s'il s'avère trop gros ;
 - ☒ Le flot d'interactions entre un diagramme de séquences détaillé et un scénario de haut niveau doivent correspondre ;
 - ☒ Supprimer le fond bleu dans Visual Paradigm.
- Vérifier que toutes les informations sont présentes dans les modèles :
 - ☒ Chaque association doit faire apparaître un nom de rôle et une multiplicité au bout de l'association ;
 - ☒ Si une association est bi-directionnelle, alors deux noms de rôle et deux multiplicités ;
 - ☒ Une association qualifiée doit faire référence à un attribut de la classe sur laquelle pointe l'association ;
 - ☒ Toutes les méthodes dans un diagramme de séquences doivent apparaître dans le diagramme de classes détaillé ;
 - ☒ Une ligne de vie dans un diagramme de séquences correspond à un objet et non une classe ;

Conserver ce paragraphe dans le rendu final et cocher les cases qui auront été réalisées/vérifiées.

Introduction

Notre équipe a pour projet de développer un outil d'organisation d'événements d'entreprise destiné aux sociétés. Son nom sera ProEventPlanner (PEP). Notre logiciel permettra de créer, modifier et suivre en temps réel des événements réalisés en entreprise comme par exemple des séminaires ou des team-building. Pour mener à bien ce projet, nous avons commencé par réaliser un dossier d'analyse et de conception regroupant un diagramme de cas d'utilisation, des scénarios nominaux et alternatifs, des diagrammes de classes et d'objets ainsi que des diagrammes de séquences. Cela nous permet ainsi de détailler les différentes fonctionnalités de notre logiciel. Nous nous sommes répartis les différentes tâches. Après avoir fait un brainstorming avec l'équipe projet au complet, afin de déterminer les fonctionnalités de notre logiciel, Léo a construit le diagramme de cas d'utilisation (*Figure 1*). Ce diagramme a permis à Émilie de faire la description des différents cas d'utilisation, à Émile et Sarah d'établir les scénarios nominaux et à Lynn de trouver les scénarios alternatifs. À partir des informations détaillées de nos fonctionnalités, Jarod a réalisé le tableau de priorisation des cas d'utilisation. Une fois cette étape finie, il nous a fallu commencer la conception grâce à des diagrammes de classes. Sarah et Émile en ont donc réalisé un simplifié (*Figure 2*) contenant uniquement les attributs des différentes classes et leurs liens. Ils ont ensuite construit des diagrammes d'objets (*Figure 3 et 4*) afin d'illustrer le fonctionnement du diagramme précédent. Ils ont ensuite construit le diagramme de classes final (*Figure 5 et 6*) contenant toutes les méthodes et constructeurs des classes décrites précédemment. Les méthodes ainsi définies ont alors permis à Jarod et Léo de réaliser respectivement les diagrammes de séquences du premier (*Figure 7*) et du deuxième (*Figure 8*) cas d'utilisation. Enfin, Émilie a rédigé et mis en page le rapport.

Expression des besoins et Analyse

Description des cas d'utilisation

Visual Paradigm Standard(ducruet)(UT2 - UPMF)

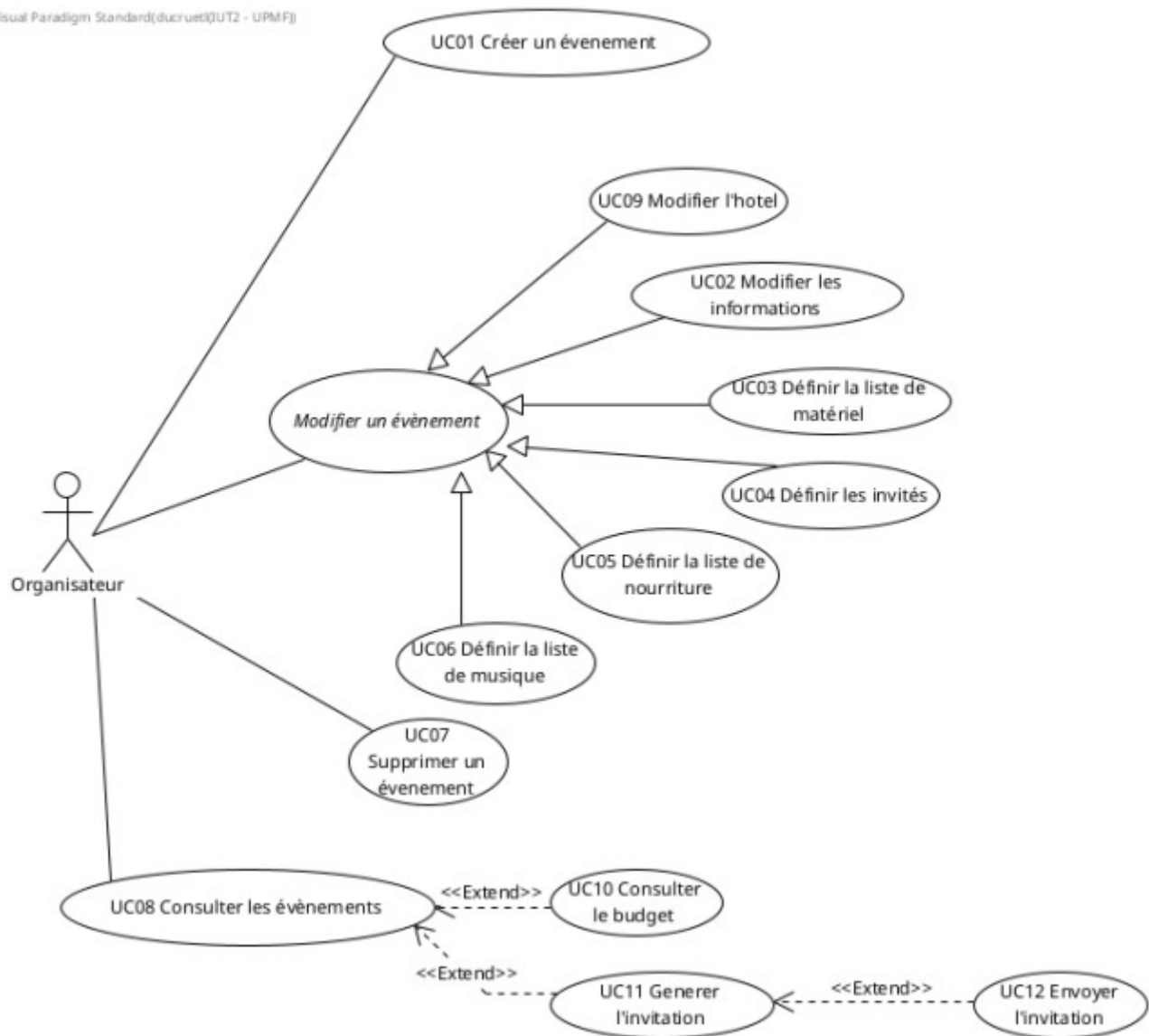


Figure 1: Diagramme des cas d'utilisation

Ce diagramme de cas d'utilisation, réalisé par Léo, permet de mieux visualiser les différentes fonctionnalités du logiciel. Nous pouvons voir que la fonctionnalité abstraite « Modifier un événement » regroupe plusieurs fonctionnalités de personnalisation de l'événement. Aussi, ce diagramme montre qu'il faut d'abord consulter les événements avant de pouvoir consulter le budget ou générer une invitation. Il en est de même pour envoyer une invitation, il est nécessaire de l'avoir générée avant.

Nom du cas d'utilisation	UC1
Libellé du cas d'utilisation	Créer un événement
Description	L'utilisateur demande à l'application de créer un événement caractérisé par : <ul style="list-style-type: none"> - le nom de l'événement - le budget de l'événement - la date et l'heure de l'événement - le lieu de l'événement qui ne peut être utilisé plusieurs fois le même jour - le type de l'événement à choisir entre plusieurs propositions fournies par l'application (ex : team-building, pot de départ...)

Nom du cas d'utilisation	UC2
Libellé du cas d'utilisation	Modifier les informations
Description	L'utilisateur peut modifier les informations de création de l'événement donc le nom, le budget, la date, l'heure, le lieu et le type en s'assurant de respecter les potentielles contraintes.

Nom du cas d'utilisation	UC03
Libellé du cas d'utilisation	Définir la liste de matériel
Description	L'utilisateur peut choisir du matériel pour sa soirée parmi ceux proposés par l'application dans la limite du budget de l'événement.

Nom du cas d'utilisation	UC04
Libellé du cas d'utilisation	Définir les invités
Description	L'utilisateur peut ajouter ou supprimer des invités qui sont caractérisés par leur nom et leur adresse mail.

Nom du cas d'utilisation	UC05
Libellé du cas d'utilisation	Définir la liste de nourriture
Description	L'utilisateur peut ajouter ou modifier la liste de plats dans la limite du budget de l'événement.

Nom du cas d'utilisation	UC06
Libellé du cas d'utilisation	Définir la liste de musiques
Description	L'utilisateur peut choisir une playlist parmi les choix de l'application. Il pourra aussi bénéficier de playlist recommandées selon le type d'événement sélectionné.

Nom du cas d'utilisation	UC07
Libellé du cas d'utilisation	Supprimer un événement
Description	L'utilisateur demande à l'application de supprimer un événement.

Nom du cas d'utilisation	UC08
Libellé du cas d'utilisation	Consulter les événements
Description	L'utilisateur demande à l'application de voir les événements qu'il a créés.

Nom du cas d'utilisation	UC09
Libellé du cas d'utilisation	Modifier l'hôtel
Description	L'utilisateur demande à l'application de modifier l'hôtel choisi uniquement si le type d'événement choisi est un séminaire

Nom du cas d'utilisation	UC10
Libellé du cas d'utilisation	Consulter le budget
Description	L'utilisateur demande à l'application de voir l'évolution du budget de l'événement en fonction des dépenses en nourriture, matériel et hôtel pour les séminaires.

Nom du cas d'utilisation	UC11
Libellé du cas d'utilisation	Générer une invitation
Description	L'utilisateur demande à l'application de visualiser une potentielle invitation pour l'événement. Celle-ci est générée par l'application.

Nom du cas d'utilisation	UC12
Libellé du cas d'utilisation	Envoyer les invitations
Description	L'utilisateur demande à l'application d'envoyer une invitation à chaque invité enregistré.

Scénarios nominaux

Créer un événement	UC1.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur fournit à la plate-forme le nom de l'événement à créer. 2. La plate-forme demande à l'utilisateur de fournir un nom, un budget, une date, un lieu et un type d'événement. 3. L'utilisateur fournit ces informations. 4. La plate-forme crée l'événement et notifie l'utilisateur.

Modifier un événement	UC2.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement à modifier. 2. La plate-forme demande à l'utilisateur de modifier les informations. 3. L'utilisateur fournit ces informations. 4. La plate-forme modifie l'événement et notifie l'utilisateur.

Définir la liste de matériel	UC3.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement à modifier. 2. La plate-forme demande à l'utilisateur de donner une liste de matériel. 3. L'utilisateur fournit cette information. 4. La plate-forme ajoute ou remplace la liste et notifie l'utilisateur.

Définir la liste d'invités	UC4.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement à modifier. 2. La plate-forme demande à l'utilisateur de donner une liste d'invités, avec leur nom et mail. 3. L'utilisateur fournit ces informations. 4. La plate-forme ajoute ou remplace la liste et notifie l'utilisateur.
Définir la liste de nourriture	UC5.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement à modifier. 2. La plate-forme demande à l'utilisateur de donner une liste de nourriture. 3. L'utilisateur fournit cette information. 4. La plate-forme ajoute ou remplace la liste et notifie l'utilisateur.
Définir la liste de musique	UC6.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement à modifier. 2. La plate-forme propose des musiques. 3. L'utilisateur coche celles qu'il souhaite. 4. La plate-forme ajoute ou remplace la liste et notifie l'utilisateur.
Supprimer un événement	UC7.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement à supprimer. 2. La plate-forme demande à l'utilisateur de confirmer son choix. 3. L'utilisateur valide la suppression. 4. La plate-forme supprime l'événement et notifie l'utilisateur.
Consulter les événements	UC8.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur consulte les événements. 2. La plate-forme affiche les informations de l'événement choisi.
Modifier l'hôtel	UC9.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit le séminaire à modifier. 2. La plate-forme demande à l'utilisateur de choisir un hôtel. 3. L'utilisateur choisit un hôtel. 4. La plate-forme sauvegarde l'hôtel et notifie l'utilisateur.
Consulter les budget	UC10.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur consulte un budget. 2. La plate-forme affiche le montant du budget.
Générer l'invitation	UC11.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement. 2. La plate-forme affiche un aperçu de l'invitation.

Envoyer l'invitation	UC12.nominal
Description	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'événement. 2. La plate-forme affiche un aperçu de l'invitation. 3. L'utilisateur appuie sur le bouton d'envoi. 4. La plate-forme envoie les mails et notifie l'utilisateur.

Scénarios alternatifs

Nom du scénario alternatif	UC1. Nom non existant
Description	Si l'utilisateur ne rentre pas de nom pour l'événement, alors la plate-forme renvoie un message d'erreur et l'invite à rentrer un nom.

Nom du scénario alternatif	UC1. Lieu non existant
Description	Si l'utilisateur ne rentre pas de lieu pour l'événement, alors la plate-forme renvoie un message d'erreur et l'invite à rentrer un lieu.

Nom du scénario alternatif	UC1. Date de début non rentrée
Description	Si l'utilisateur ne rentre pas de date de début pour l'événement, alors la plate-forme renvoie un message d'erreur et l'invite à rentrer une date de début.

Nom du scénario alternatif	UC1. Budget non existant
Description	Si l'utilisateur ne rentre pas de budget, alors le budget sera automatiquement de 0.

Nom du scénario alternatif	UC1. Événement déjà existant
Description	Si lors de la création, l'utilisateur utilise un nom déjà utilisé par un autre événement, alors la plate-forme renvoie un message d'erreur et l'invite à changer de nom.

Nom du scénario alternatif	UC1. Événement à la même date et au même lieu
Description	Si lors de la création, l'utilisateur rentre un lieu déjà utilisé par un autre événement à la même date, alors la plate-forme renvoie une erreur et invite l'utilisateur à changer soit la date, soit le lieu, ou les deux.

Nom du scénario alternatif	UC1. Autre type que double dans le budget
Description	Si l'utilisateur tente de rentrer un élément de type différent de double dans le label budget, alors le label ne se remplira pas.

Nom du scénario alternatif	UC2. Lieu non existant
Description	Si l'utilisateur efface le lieu et ne rentre pas de nouveau lieu, alors la plate-forme renvoie un message d'erreur et le lieu reste inchangé.

Nom du scénario alternatif	UC1. Événement déjà existant
Description	Si lors de la modification, l'utilisateur change le nom de l'événement pour un nom déjà existant, alors la plate-forme renvoie un message d'erreur et le nom reste inchangé.

Nom du scénario alternatif	UC2. Événement à la même date et au même lieu
Description	Si lors de la modification, la date ou le lieu change et que le lieu est déjà utilisé par un autre événement à la même date, alors la plateforme renvoie une erreur et invite l'utilisateur à changer soit la date, soit le lieu, ou les deux.
Nom du scénario alternatif	UC2.Budget non existant
Description	Si l'utilisateur efface le budget de l'événement et ne rentre pas de nouveau budget, alors le budget prend automatiquement la valeur de 0.
Nom du scénario alternatif	UC2.Autre type que double dans le budget
Description	Si l'utilisateur tente de rentrer un élément de type différent de double dans le label budget, alors le label ne se remplira pas.
Nom du scénario alternatif	UC3.Pas assez de budget pour du matériel
Description	Si l'utilisateur souhaite réserver du matériel mais que son budget n'est pas assez élevé, alors la plateforme renvoie un message d'erreur « Budget dépassé » et ne prend pas en compte l'ajout de matériel.
Nom du scénario alternatif	UC4. Adresse mail non rentrée
Description	Si l'utilisateur ne rentre pas d'adresse mail pour l'invité alors la plateforme renvoie un message d'erreur et lui demande de rentrer une adresse mail.
Nom du scénario alternatif	UC4. Nom non rentré
Description	Si l'utilisateur ne rentre pas de nom pour l'invité, alors la plateforme renvoie un message d'erreur et l'invite à rentrer un nom.
Nom du scénario alternatif	UC4. Invité en double
Description	Si l'utilisateur rentre deux invités ayant la même adresse mail, alors la plateforme renvoie un message d'erreur et invite l'utilisateur à supprimer ou modifier le second invité.
Nom du scénario alternatif	UC6.Pas assez de budget pour la nourriture
Description	Si l'utilisateur souhaite acheter de la nourriture mais que son budget n'est pas assez élevé, alors la plateforme renvoie un message d'erreur « Budget dépassé » et ne prend pas en compte l'ajout de nourriture.
Nom du scénario alternatif	UC9.Pas assez de budget pour l'hôtel
Description	Si l'utilisateur souhaite réserver un hôtel mais que son budget n'est pas assez élevé, alors la plateforme renvoie un message d'erreur « Budget dépassé » et ne prend pas en compte la réservation de l'hôtel.

Nom du scénario alternatif	UC12.Pas d'invité
Description	Si l'utilisateur souhaite envoyer son invitation (déjà visualisée) mais qu'il n'y a pas d'invité dans la liste d'invités, alors la plate-forme renvoie un message d'erreur indiquant qu'il n'y a aucun invité à qui envoyer l'invitation.

Priorisation des cas d'utilisation

Priorité	Cas d'utilisation	Justification
1	UC1	Créer un événement est la base de notre logiciel.
	UC2	Modifier les informations dite « importantes » telles que le nom, la date, le lieu et le type.
2	UC3	L'organisateur peut définir une liste de matériel nécessaire en fonction du budget.
	UC7	Supprimer un événement quand il est fini est important pour éviter un surnombre d'événements créés et que l'on ne sait pas s'ils ont eu lieu ou non.
	UC8	Consulter les événements peut s'avérer utile lorsque que l'on crée plusieurs événements.
	UC9	Modifier l'hôtel peut s'avérer important car il faut un lieu spécifique pour l'événement.
3	UC4	Définir des invités est utile mais il n'est pas obligatoire de le faire via notre logiciel.
	UC5	Définir une liste de nourriture peut s'avérer utile pour avoir un aperçu du budget dépensé dans cette catégorie.
	UC6	On peut générer une invitation personnalisée pour les invités. Définir une liste de musiques est utile mais pas obligatoire surtout si un DJ est invité.
	UC11	On peut générer une invitation personnalisée pour les invités.
	UC12	On peut envoyer l'invitation à toutes les personnes qui sont dans la liste des invités.

Architecture proposée

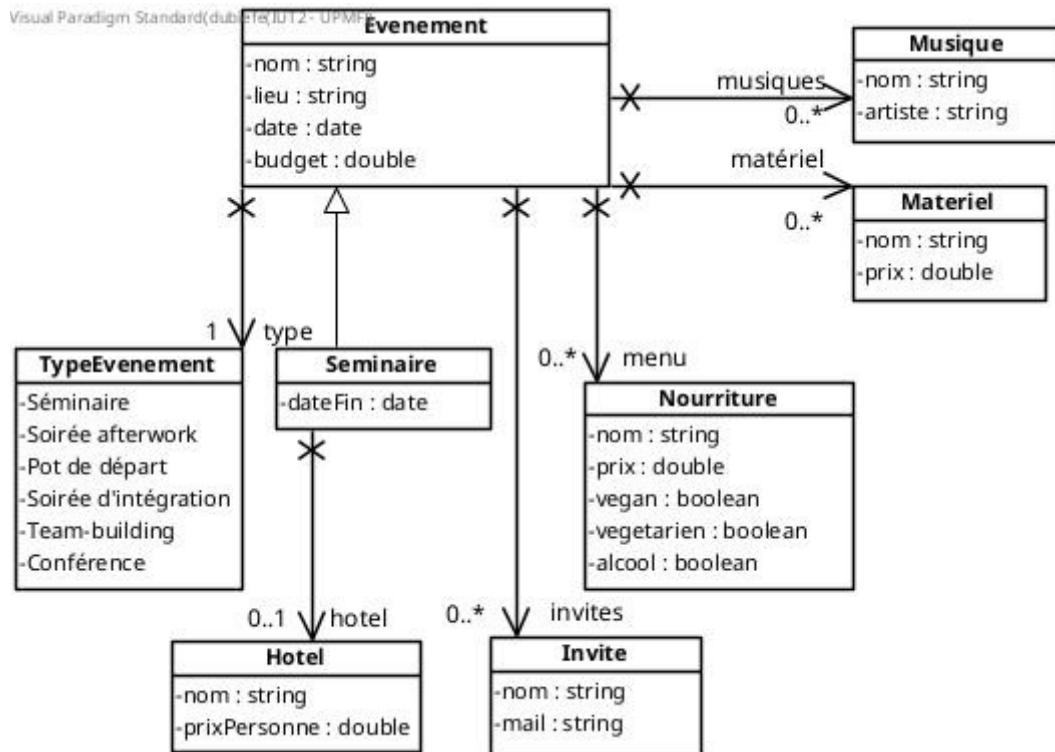


Figure 2: Diagramme de classes simplifié

Ce diagramme de classe, réalisé par Sarah et Émile, est simplifié de manière à ne faire apparaître que les attributs des différentes classes et les associations. Nous pouvons ainsi mieux voir les éléments obligatoires à la création d'un événement puisqu'ils ont tous une cardinalité de 1 (nom, lieu, date, budget, et type). Dans la suite logique, les autres attributs sont donc les données modifiables en plus après la création. Nous pouvons également constater que le type d'événement « séminaire » est défini autrement, puisqu'il peut durer plusieurs jours. Une date de fin et un hôtel sont donc requis. Cependant, l'hôtel n'étant pas nécessaire à la création d'un séminaire, il n'a pas une cardinalité de 1, et peut donc être modifié en amont. Enfin, la classe « TypeEvenement » est une classe enum qui a pour valeur les six types d'événements disponibles dans le logiciel.

Exemples d'utilisation

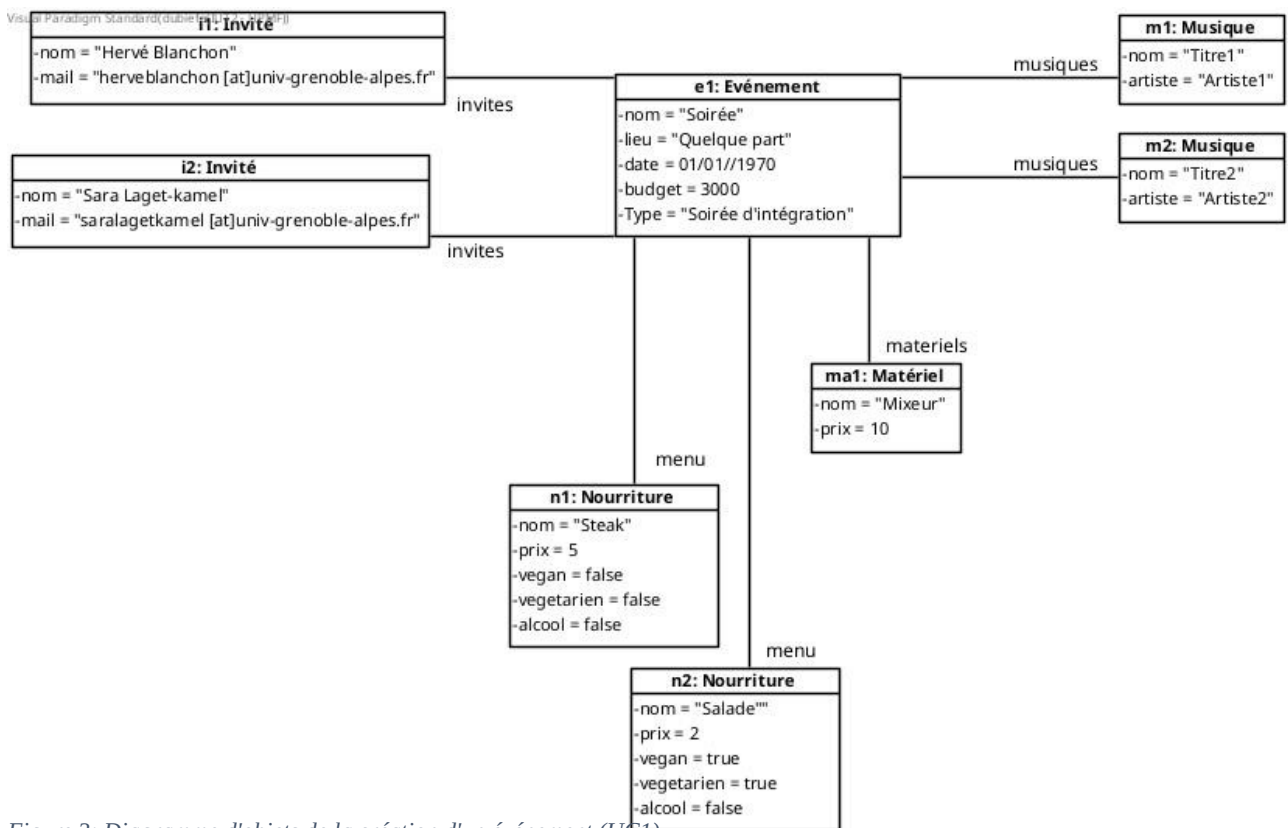


Figure 3: Diagramme d'objets de la création d'un événement (UC1)

Ce diagramme d'objets, réalisé par Sarah et Émile, illustre la création d'un événement dont les caractéristiques ont été définies. Il est également composé de deux musiques, d'un menu de deux plats et d'une liste d'invités de deux membres.

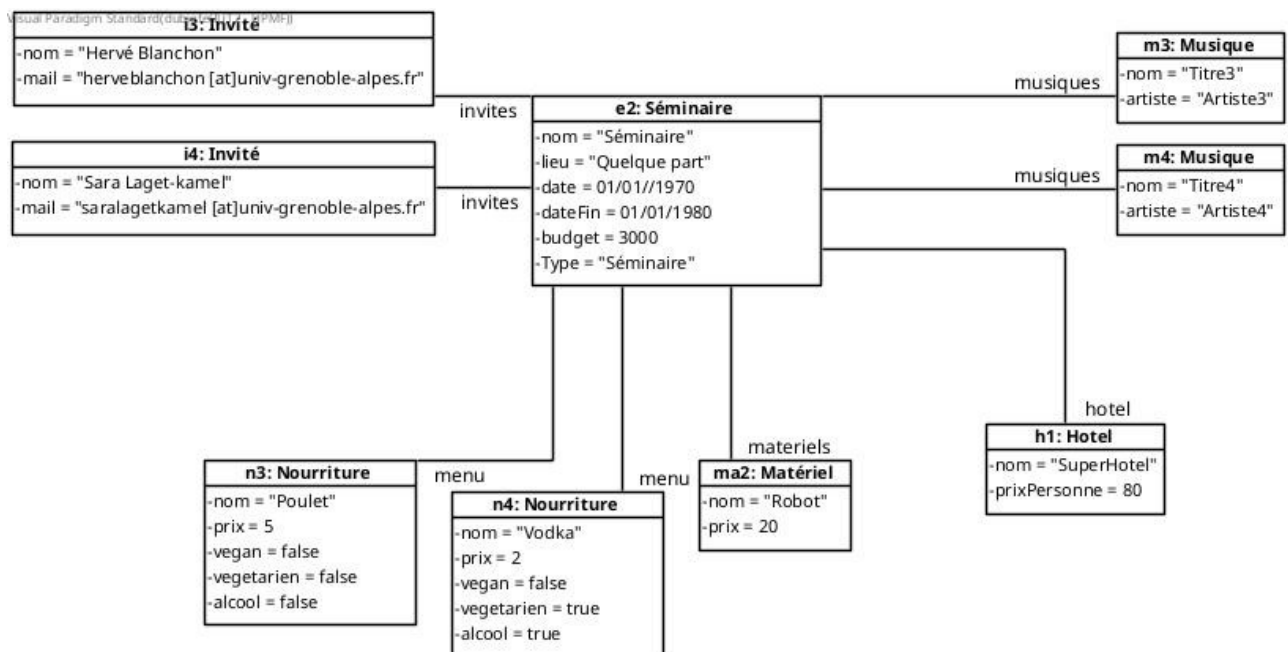


Figure 4: Diagramme d'objets de la création d'un séminaire (UC1)

Ce diagramme d'objets, réalisé par Sarah et Émile, illustre la création d'un séminaire dont les caractéristiques ont été définies. Il est également composé de deux musiques, d'un menu de deux plats et d'une liste d'invités de deux membres. Il dispose en plus d'un hôtel.

Conception détaillée

Architecture détaillée

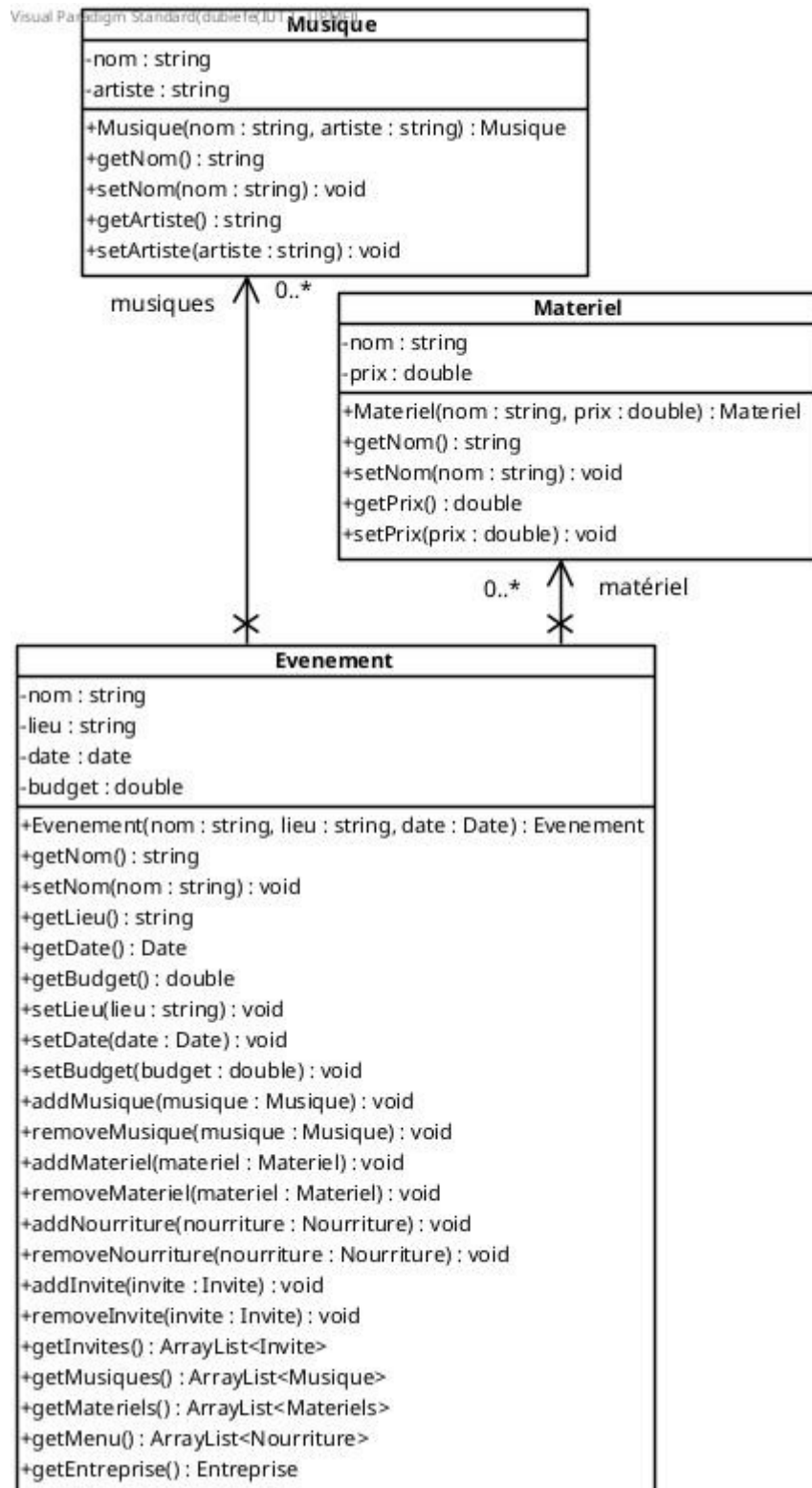


Figure 5: Diagramme de classes détaillé (Partie 1)

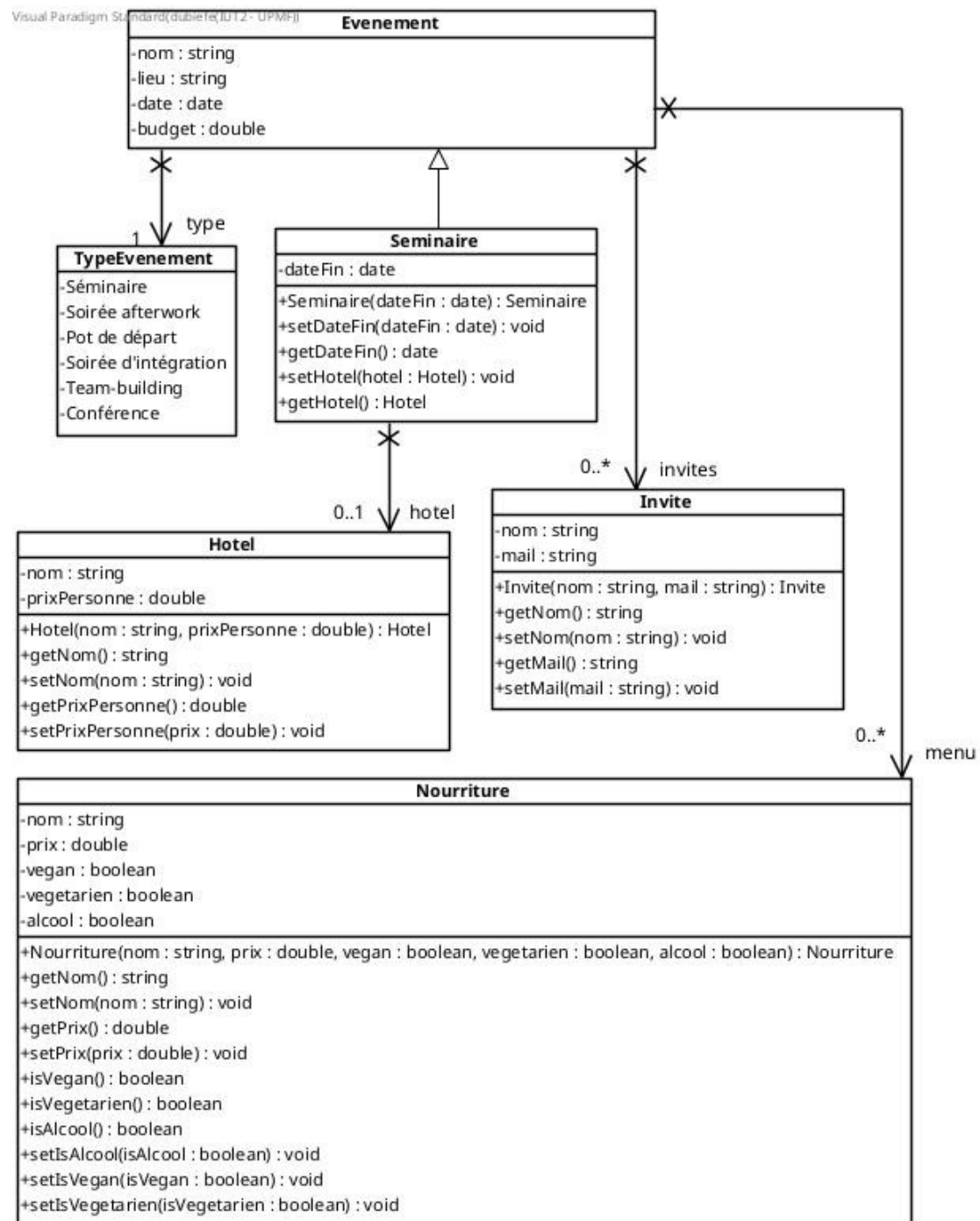


Figure 6: Diagramme de classes détaillé (Partie2)

Ce diagramme de classes détaillé, réalisé par Sarah et Émile, vient compléter le diagramme de classe simplifié fait en amont. En effet, les méthodes et constructeurs de chaque classe sont détaillés. Le diagramme a été découpé en deux puisqu'il était trop conséquent. La classe « Evenement » a donc été écrite dans son entièreté dans la partie 1. Dans la deuxième partie, cette classe a été réduite pour des raisons de clarté du diagramme.

Détail des scénarios

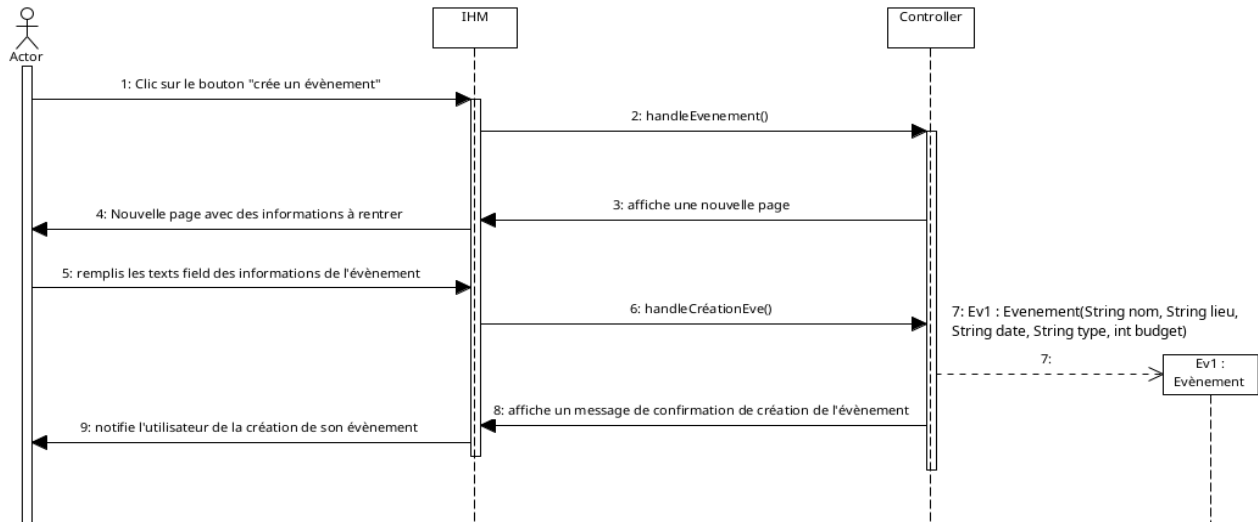


Figure 7: Diagramme de séquences de la création d'un événement (UC1)

Ce diagramme de séquences, réalisé par Jarod, illustre le fonctionnement du cas d'utilisation de priorité 1 : créer un événement (UC1). Il est composé de huit étapes et entraîne la création d'un objet de type événement à l'étape 7.

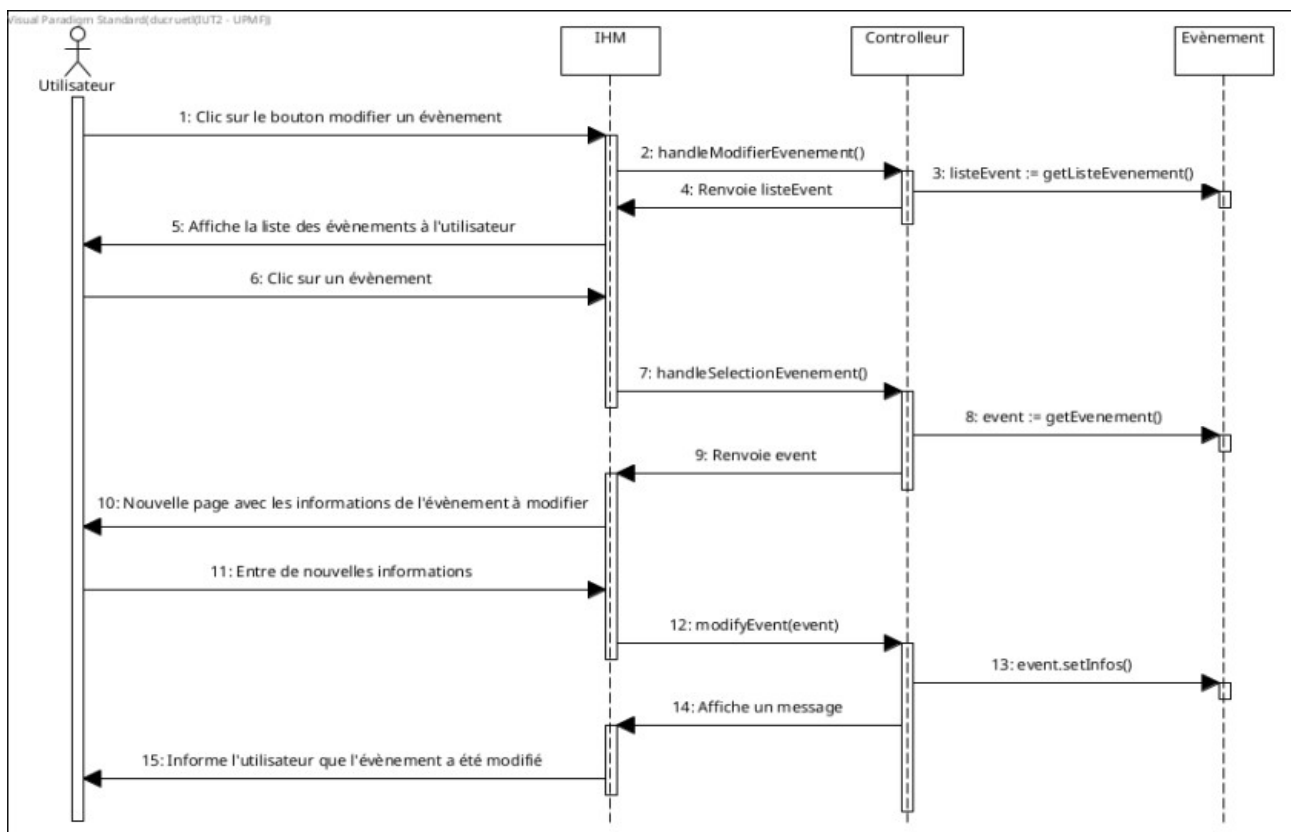


Figure 8: Diagramme de séquences de la modification d'un événement (UC2)

Ce diagramme de séquences, réalisé par Léo, illustre le fonctionnement du cas d'utilisation de priorité 1 : modifier un événement (UC2). Il est composé de 15 étapes.

Conclusion

Après avoir réalisé ce dossier d'analyse et de conception, nous avons pu affiner notre projet en établissant ses fonctionnalités. Cette tâche n'a pas été aisée au début, puisqu'il a fallu se mettre d'accord sur les fonctionnalités à planter. Nous devons prendre en compte le temps disponible mais aussi nos capacités respectives de codage. Nous avons, dans un premier temps, défini trop peu de fonctionnalités. C'est pourquoi nous en avons rajoutées, ce qui a entraîné une refonte des tableaux réalisés en première partie. Après ce contre-temps, la construction des diagrammes de classes et de séquences ne nous a pas posé de problèmes majeurs. Ainsi, nous avons pu finaliser les différentes fonctionnalités de notre projet. Notre logiciel aura pour but de dresser un planning des événements d'entreprise organisés par l'utilisateur. Il permettra de créer, modifier et consulter des événements à venir. De plus, il détaillera avec une grande précision la création et la modification des événements en raison des nombreux attributs disponibles. Il sera également possible de consulter le budget de chaque événement en temps réel afin de garantir une bonne visualisation pour l'utilisateur. Enfin, le logiciel offre la possibilité de générer une invitation type et de l'envoyer par mail à tous les invités renseignés dans l'événement. Toutes les étapes réalisées dans ce dossier nous ont permis d'établir ces fonctionnalités, leur fonctionnement mais aussi les contraintes qu'elles peuvent engendrer. Cela nous sera donc très utile lors de l'étape de réalisation du projet au cours de laquelle nous coderons le logiciel. En effet, avoir réalisé les diagrammes de classes et de séquences en amont nous permettra d'être plus efficaces et de savoir immédiatement ce que nous devons coder. Il sera aussi plus facile de partitionner la tâche de codage puisqu'il nous suffira de suivre ce que nous avons défini dans ce dossier.