

Interface et Multimédia
IFT-215

Travail Pratique 4

Amandine Fouillet - 14 130 638
Frank Chassing - 14 153 710
Laurent Sénécal-Léonard - 14 143 484

9 mars 2015

Table des matières

Introduction	4
1 Modélisation de la gestion de la Communication	4
2 Modélisation de la gestion des Photos	5
3 Modélisation de la gestion du Calendrier	6
4 Modélisation de la gestion des Dépenses	7
5 Modélisation de la gestion de la Carte	8
6 Modélisation de la gestion des Contacts	9
7 Diagramme de classe global	10
Conclusion	11

Introduction

Après avoir étudié les différents acteurs, analysé leurs différents environnements et détaillé l'ensemble des fonctionnalités que nous voulons voir apparaître dans notre application, nous exposerons dans ce rapport l'ensemble des modélisations de nos données par le biais de diagrammes UML. Pour cela, nous commencerons par décrire les diagrammes de nos fonctionnalités en identifiant les différents attributs et les différentes méthodes liés à celles-ci. Puis nous effectuerons un compte rendu global de notre diagramme afin d'observer les différents liens entre nos classes.

1 Modélisation de la gestion de la Communication

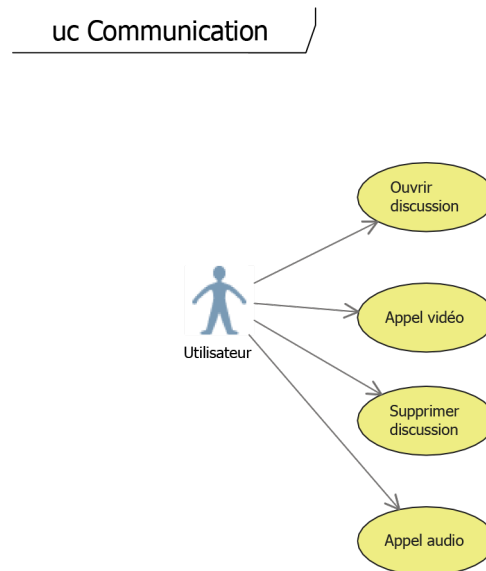


FIGURE 1 – Cas d'utilisation communication

Tout d'abord, sur le diagramme de cas d'utilisation de la gestion de la communication (figure 1), il est indiqué les fonctionnalités que peut exécuter l'utilisateur. Dans ce présent, celui-ci peut ouvrir une discussion, faire un appel vidéo, supprimer une discussion et faire un appel audio. Au

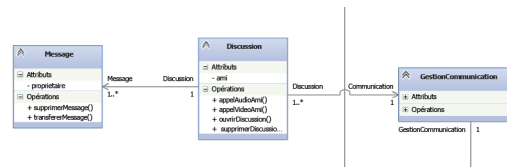


FIGURE 2 – Diagramme de classe communication

niveau du diagramme de classe (figure 2), on y trouve trois classes distinctes. La classe Discussion sert à modéliser une discussion avec un ami et possède quatre méthodes permettant l'utilisation des fonctionnalités. Ensuite, une classe Message a comme utilité de modéliser un message. Celle a comme attribut le propriétaire et possède les méthodes de suppression et de transfert de messages.

2 Modélisation de la gestion des Photos

uc Photos

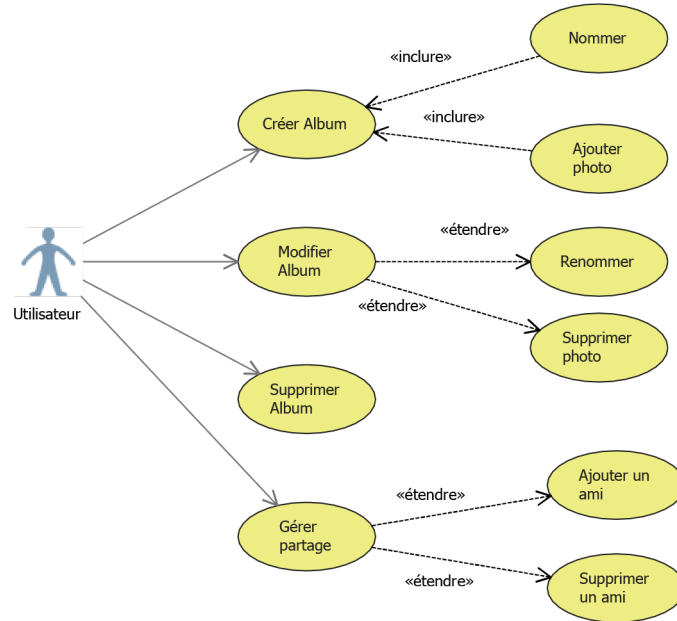


FIGURE 3 – Cas d'utilisation photos

Pour commencer, le diagramme de cas d'utilisation de la gestion des photos (figure 3) permet de montrer explicitement les fonctionnalités que peut exécuter l'utilisateur. En effet, il pourra créer un album, le modifier, le supprimer et gérer un partage des photos de ses albums à un ami. Au

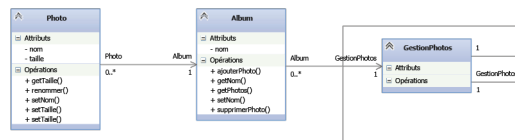


FIGURE 4 – Diagramme de classe photos

niveau du diagramme de classe (figure 4), celui-ci contient une classe Photo contenant le nom et la taille de la photo comme attribut. Concernant les méthodes, on y trouve des accesseurs (get et set) et une méthode pour renommer la photo. Ensuite, une classe Album possède comme attribut le nom de l'album, des méthodes accesseurs et deux méthodes pour ajouter ou supprimer un album. Finalement, une classe pour la gestion des photos permet de gérer les albums et les partages avec des amis.

3 Modélisation de la gestion du Calendrier

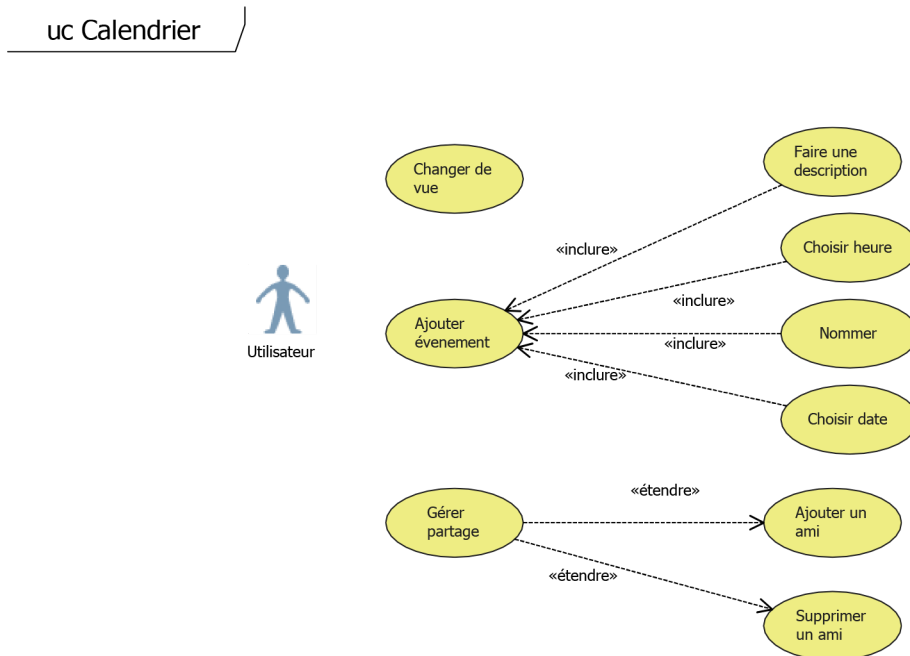


FIGURE 5 – Cas d'utilisation calendrier

Le diagramme de cas d'utilisation du calendrier (figure 5) montre de manière explicite les différentes fonctionnalités que peut exécuter un utilisateur. En effet, ce dernier peut ajouter un évènement tout en faisant une description, assigner un nom, choisir une heure et une date. De plus, il est possible de changer l'affichage du calendrier et de gérer les partages avec ses amis. Au niveau du diagramme

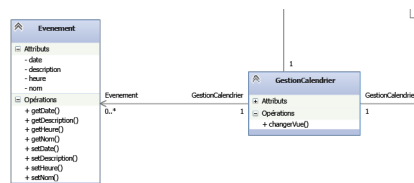


FIGURE 6 – Diagramme de classe calendrier

de classe (figure 6), la classe évènement a comme attribut privé la date, la description, l'heure et le nom. Celui-ci possède plusieurs accesseurs afin de pouvoir récupérer ou modifier les attributs. Ensuite, une classe pour la gestion du calendrier offre comme méthode de changer l'affichage du calendrier.

4 Modélisation de la gestion des Dépenses

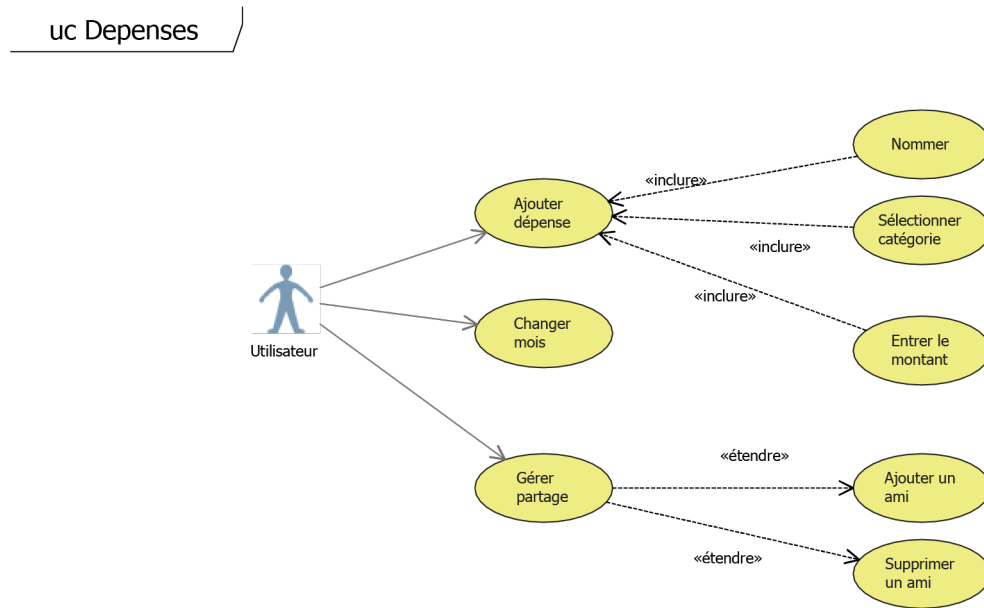


FIGURE 7 – Cas d'utilisation dépenses

Le diagramme de cas d'utilisation de la gestion des dépenses (figure 7) montre les trois fonctionnalités que peut exécuter un utilisateur. En effet, il peut ajouter une dépense en incluant un nom de dépense, une catégorie et le montant total. De plus, celui-ci peut changer le mois à lequel il assigne une dépense et il peut gérer le partage de ses informations de dépenses avec un ami. [!h] Au niveau

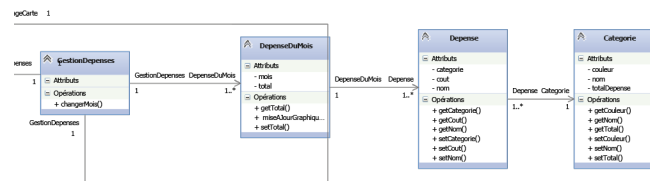


FIGURE 8 – Diagramme de classe dépenses

du diagramme de classe de la gestion des dépenses, on y trouve une classe GestionDépenses ayant comme méthode publique de changer le mois. Celle-ci est reliée à une classe DepenseDuMois permettant de calculer le total de la dépense du mois. Ensuite, une classe Dépense servant à attribuer les informations des dépenses et possédant plusieurs méthodes publiques d'accesseurs. Finalement, on trouve une classe gérant les catégories de dépenses entrées par l'utilisateur. Celle-ci détient comme attribut privé la couleur pour l'affichage de la dépense, le nom de la catégorie et le total de la dépense et cette dernière à plusieurs méthodes publiques d'accesseur pour aller chercher ou mettre à jour des attributs.

5 Modélisation de la gestion de la Carte

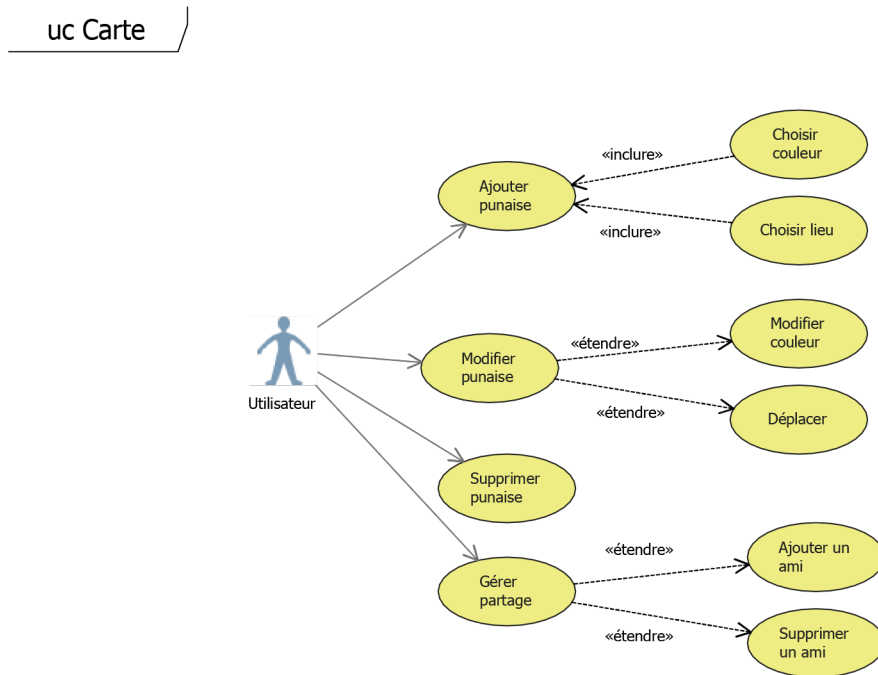


FIGURE 9 – Cas d'utilisation carte

Le diagramme de cas d'utilisation (FIGURE 9) permet de nous rappeler les différentes fonctionnalités que peut exécuter l'utilisateur dans cette section. Celui-ci pourra à tout moment ajouter une punaise sur la carte du monde afin de dire qu'il a voyagé à l'endroit où se situe la punaise. Il pourra bien évidemment déplacer ou bien supprimer sa punaise. Il pourra également modifier la couleur de celle-ci pour créer des groupes ou des jeux de couleur. Au niveau du diagramme de

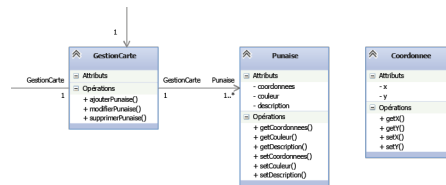


FIGURE 10 – Diagramme de classe carte

classe (FIGURE 10), nous aurons une classe permettant la gestion de la carte qui contiendra une ou plusieurs punaises ainsi que les différentes actions sur les punaises (ajouter, modifier, supprimer). La classe Punaise, quant à elle, permet de modéliser une punaise. Elle possède donc comme attribut les coordonnées de la punaise, sa couleur et une description permettant à l'utilisateur de raconter les joies de ses voyages. En ce qui concerne les méthodes, nous retrouvons les différents accesseurs (get et set) permettant de récupérer ou modifier les attributs. Nous avons également ajouté une classe Coordonnées avec une position x et y en attribut représentant respectivement l'abscisse et l'ordonnée.

6 Modélisation de la gestion des Contacts

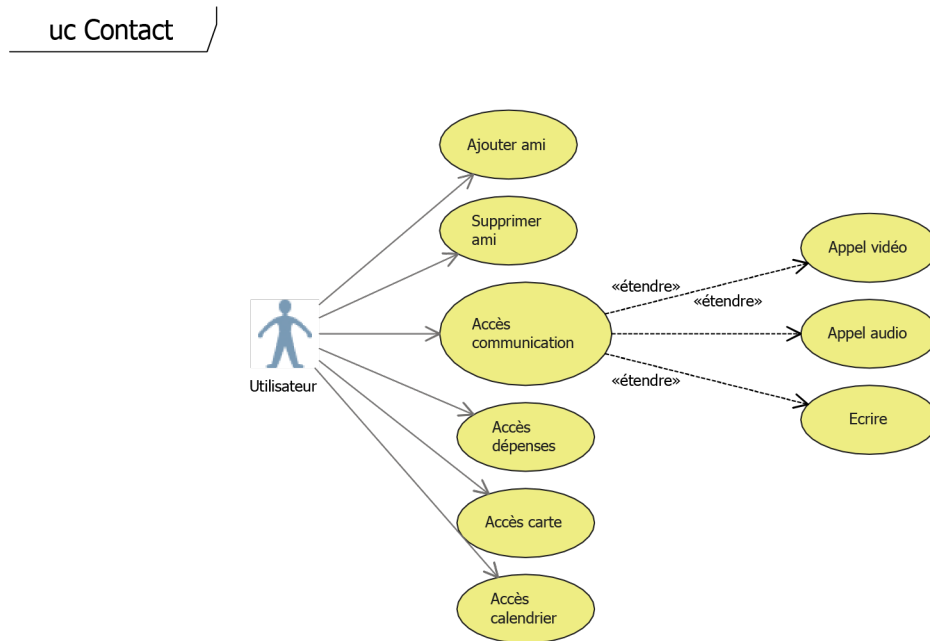


FIGURE 11 – Cas d'utilisation contacts

Nous retrouvons encore ici un diagramme de cas d'utilisation (FIGURE 11) modélisant les fonctionnalités de la section Contacts. L'utilisateur sera en mesure d'ajouter ou bien supprimer un contact à tout moment. Il pourra également observer les différentes informations sur le contact telles que ses dépenses, sa carte du monde, ses photos ou son emploi du temps. L'utilisateur aura également un accès pour communiquer avec le contact soit par vidéo, audio ou chat. La modélisation des données

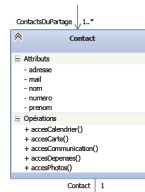


FIGURE 12 – Diagramme de classe contact

est représentée par une seule classe « Contact » (FIGURE 12). Chaque contact sera un objet de cette classe et possèdera des paramètres correspondant à ses informations personnelles telles que son nom, son prénom, son numéro, son mail ou encore son adresse. Pour chaque contact, l'utilisateur a un accès pour chaque information de celui-ci telle que son calendrier, sa carte ou son emploi du temps et un accès pour communiquer avec lui. C'est pourquoi nous retrouvons les différentes méthodes d'accès : `accesCalendrier()`, `accesCarte()`, `accesCommunication()`, `accesDepense()` et `accesPhoto()`. Ces méthodes donneront l'accès à l'utilisateur aux différentes sections. L'utilisateur aura également la possibilité de supprimer le contact grâce à la méthode `supprimerContact()`.

7 Diagramme de classe global

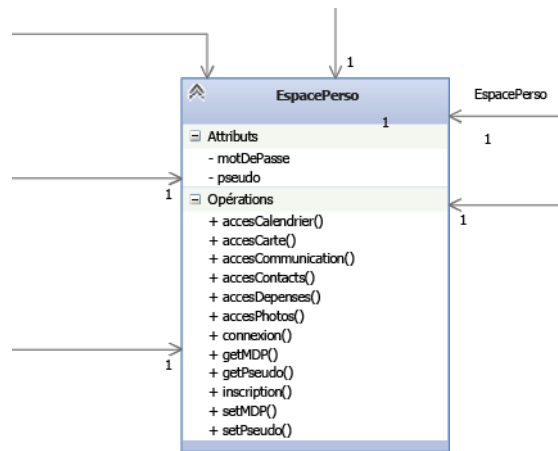


FIGURE 13 – Diagramme de classe espace personnel

La classe **EspacePerso** (FIGURE 13) est la classe qui va permettre de gérer le compte utilisateur ainsi que toutes ses données. Cette classe a deux attributs privés : le pseudo et le mot de passe qui permettent de retrouver et reconnaître l'utilisateur. Elle contient des méthodes d'accès (get et set) pour récupérer ou modifier ses attributs. Les méthodes publiques `inscription()` et `connexion()` permettent respectivement de gérer la création de l'espace personnel et l'accès à cet espace. On retrouve ensuite toutes les méthodes d'accès aux différentes données de l'utilisateur : ses contacts, sa carte, son calendrier, ses dépenses, ses communications et ses photos.

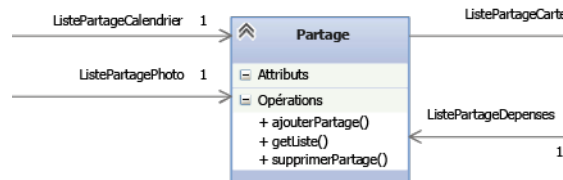


FIGURE 14 – Diagramme de classe partage

La classe **Partage** (FIGURE 14) est la classe qui va permettre à l'utilisateur de gérer les listes de contacts avec qui il veut partager ses différentes informations. Pour rappel, dans notre application, il sera possible à l'utilisateur de partager ses photos, son calendrier, sa carte ainsi que ses dépenses. Il nous semble important de laisser la possibilité à l'utilisateur de créer des listes de partage différentes afin de lui laisser la possibilité de ne pas tout partager aux mêmes personnes. Par exemple un étudiant peut décider de partager ses dépenses seulement avec ses parents et ses photos de soirée seulement avec ses amis. Pour implémenter cela, nous avons décidé de créer la classe **Partage** qui sera finalement qu'une sorte de liste améliorée de contacts. Pour chacune des catégories (carte, dépenses, photos et calendrier) un partage sera créé et grâce aux méthodes `ajouterPartage()` et `supprimerPartage()` il sera possible d'ajouter ou de supprimer des contacts à qui l'accès à la catégorie sera autorisé.

La FIGURE 15 représente le diagramme de classe de l'application dans sa globalité. On retrouve les différentes classes que l'on a détaillées et expliquées précédemment. On peut cependant constater les dépendances et les relations entre ces différentes classes :

- ## Conclusion

11

Table des figures

1	Cas d'utilisation communication	4
2	Diagramme de classe communication	4
3	Cas d'utilisation photos	5
4	Diagramme de classe photos	5
5	Cas d'utilisation calendrier	6
6	Diagramme de classe calendrier	6
7	Cas d'utilisation dépenses	7
8	Diagramme de classe dépenses	7
9	Cas d'utilisation carte	8
10	Diagramme de classe carte	8
11	Cas d'utilisation contacts	9
12	Diagramme de classe contact	9
13	Diagramme de classe espace personnel	10
14	Diagramme de classe partage	10
15	Diagramme de classe	11