

Rapport d'architecture

Remise des diplômes



Polytech Nice-Sophia
Groupe I

Table des matières

Introduction	2
Vue fonctionnelle	2
Description des acteurs	2
Scénarios	2
Diagramme de cas d'utilisations	4
Diagramme de composants	5
Interfaces exposées	5
Vue développement	8
Diagramme de classe des objets métiers	8
Modèle relationnel de stockage	9
Justification du mapping objet-relationnel	9
Conclusion	10

Introduction

Poly'Diploma est un logiciel visant à simplifier l'organisation de la cérémonie de remise des diplômes. Il doit automatiser certaines tâches afin de réduire les erreurs et de faciliter l'organisation de l'événement.

I. Vue fonctionnelle

A. Description des acteurs

Nous avons identifié quatre acteurs pouvant interagir avec notre système RDD (Remise Des Diplômes) :

- **Service communication** : Cet acteur va se charger de fixer la date de la cérémonie pour ensuite lancer les invitations aux étudiants et aux VIP depuis le système. Il va aussi signaler au système si un VIP est absent. Enfin, le service communication va créer des offres d'intermède pour les associations et va pouvoir valider la réponse la plus intéressante
- **Service comptable** : Cette personne va se charger de payer les prestataires et de gérer les justificatifs de paiement sur le système.
- **Étudiant** : Étudiant de Polytech, il est invité à la cérémonie s'il a validé son diplôme. il va pouvoir se connecter au système RDD afin de valider sa présence. Lors de cette validation, l'étudiant pourra inscrire jusqu'à deux personnes gratuitement. Au-delà de deux personnes, l'étudiant devra payer pour chaque accompagnateur supplémentaire.
- **Association** : Une association pouvant présenter des animations (association pom-pom girl, club musique...etc). Elle répond à un des appels d'intermède lancé par le système de communication.
- **HyperPlanning**: Acteur externe, le gestionnaire de notes
- **Service d'impression de l'université**: Acteur externe, imprime les diplômes, donne une date à laquelle tous les diplômes seront imprimés
- **Banque**: Acteur externe, reçoit et demande les paiements
- **Serveur SMTP**: Acteur externe, permet de recevoir et envoyer des mails aux utilisateurs.
- **Centre des congrès**: Acteur externe, sous la forme d'un service externe, prend acte de la réservation de salle pour l'événement

B. Scénarios

Scénario "Le service communication choisit une date pour la cérémonie"

- Après avoir obtenu la liste des étudiants diplômés, le service d'impression de l'université donne la date d'impression des diplômes
- Le service communication fixe une date pour la cérémonie, ultérieure à cette date d'impression

Scénario "Le service comptable paie les factures"

- Le service comptable réserve une salle pour la cérémonie

- Le centre des congrès envoie sa facture au service comptable
- Le service comptable paie la facture et ajoute le justificatif de paiement

Scénario “Étudiant accepte l’invitation à la cérémonie et invite 3 personnes”

- L’étudiant reçoit une invitation
- Il accepte l’invitation sur l’application
- Il indique le nombre de personnes qu’il souhaite inviter sur l’application
- Il invite gratuitement 2 personnes
- Il paie pour la 3ème personne
- 4 personnes sont ajoutées au nombre total d’invités (étudiant + 3 accompagnateurs) et l’étudiant est marqué comme accepté

Scénario “Étudiant refuse l’invitation”

- L’étudiant reçoit une invitation
- Il refuse l’invitation sur l’application
- Il n’est pas ajouté au nombre total d’invités et l’étudiant est marqué absent avant la deadline

Scénario “Étudiant ne répond pas”

- L’étudiant reçoit une invitation
- La date limite est atteinte donc il n’est pas ajouté au nombre total d’invités et l’étudiant est marqué absent

Scénario “Association répond à un appel d’offre”

- le service communication fait un appel d’offre pour un intermède de 10 minutes
- Une association se connecte sur l’interface des associations et propose une offre pour cet intermède (Exemple : {“description”: “PomPom girl”, “nombre Participants”: 20})
- le service de communication accepte cette offre, un message est envoyé pour l’association et l’intermède est inséré dans le planning, l’appel d’offre est fermé.

Scénario “PDG ne vient pas”

- Le PDG reçoit une invitation
- Il refuse l’invitation sur l’application/par mail
- Il est marqué comme absent à la cérémonie

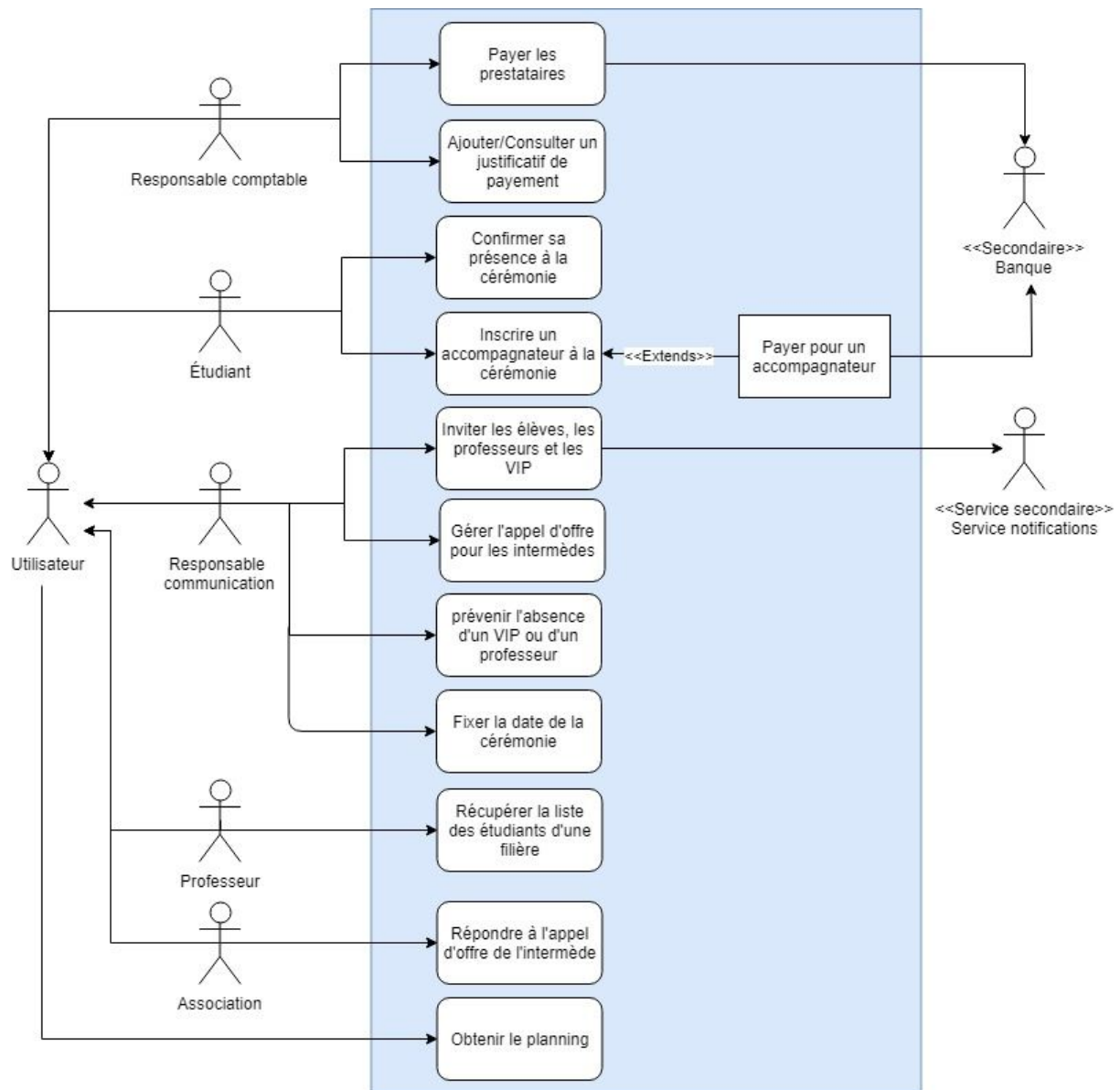
Scénario “réserver une salle au centre des congrès”

- Après la date limite pour répondre aux invitations, le centre des congrès est contacté pour réserver une salle pouvant contenir au moins tous les participants, à la date fixée pour la durée calculée.
- Le centre des congrès répond en donnant la salle, sa capacité et son prix pour la durée de l’évènement.

Scénario “Un participant veut voir le planning”

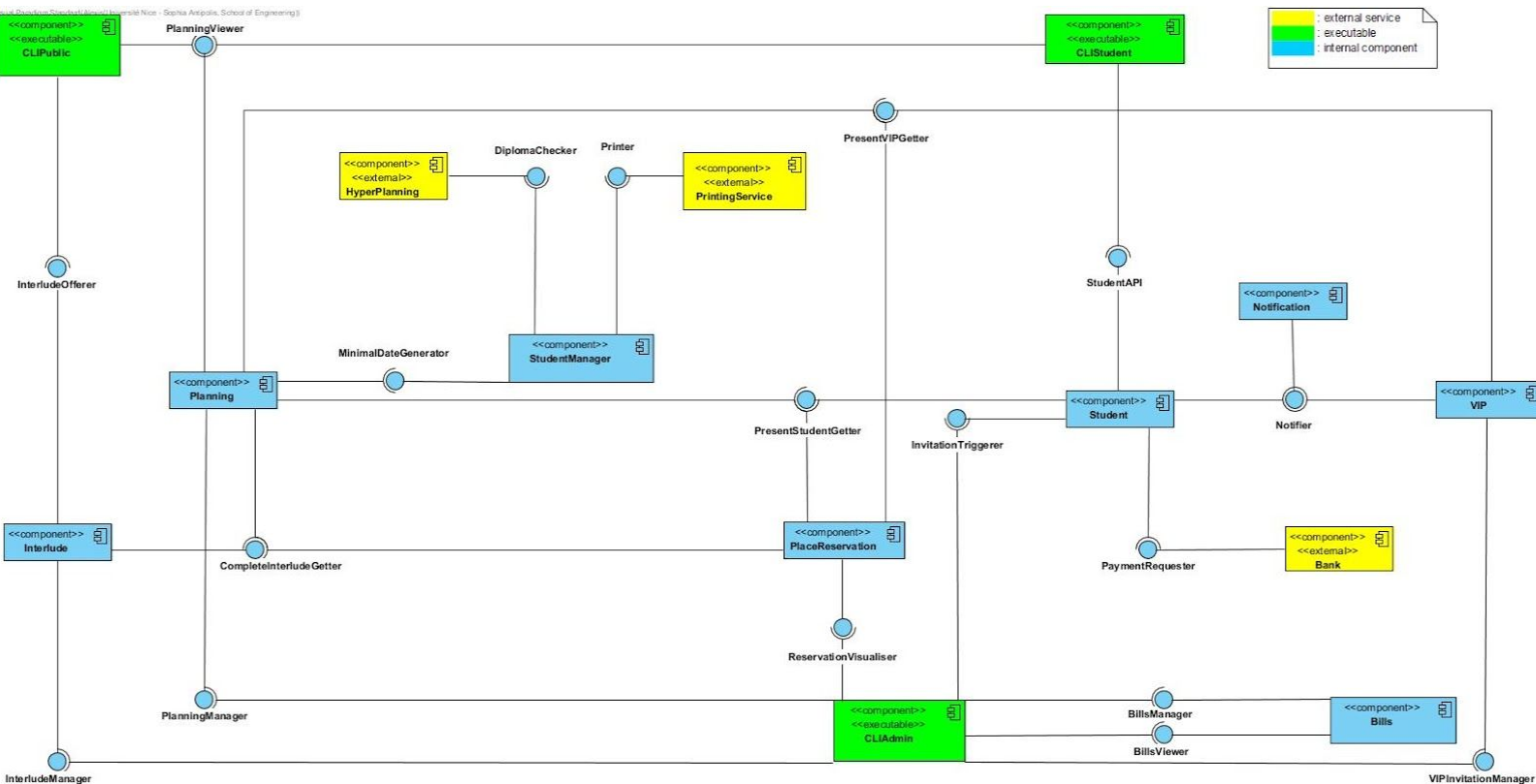
- Il se connecte sur l’application
- Il clique sur le bouton “Voir le planning”

C. Diagramme de cas d'utilisations



Le cas d'utilisation "Obtenir le planning" permet à la fois d'obtenir les horaires de passage des différentes filières et associations et d'obtenir les horaires de passage des VIP. Le cas d'utilisation "Fixer la date de la cérémonie" définit automatiquement la deadline pour les inscriptions des associations.

D. Diagramme de composants



E. Déroulement du diagramme de composants

1. Le responsable communication va accéder à l'interface "PlanningManager" depuis "CLIadmin", afin de créer une nouvelle cérémonie pour l'année 2018-2019 dans le component "Planning".
2. Le component "Planning" va passer par l'interface "MinimalDateGenerator", afin de récupérer la date avant laquelle on ne peut rien organiser. En effet, le component "StudentManager" va demander la liste des diplômés au service externe "HyperPlanning" par l'interface "DiplomaChecker".
3. Lorsque le component "StudentManager" récupère la liste des diplômés, il peut envoyer cette liste au service externe "PrintingService" en passant par l'interface "askPrint", afin d'imprimer les diplômes. Le service externe renvoie directement la date de livraison des diplômes.
4. Le responsable communication va accéder à "VIPInvitationManager" depuis "CLIadmin" pour importer la liste des VIP et des professeurs qui sont supposé participer à la cérémonie et saisir tous les "VIP".
5. Le responsable communication va échanger avec le responsable du centre des congrès afin de trouver une date se trouvant après la date de livraison des diplômes et un salle adéquat à l'estimation du nombre de personne (Étudiant + VIP + professeur).
6. Ensuite, il va accéder à l'interface ReservationVisualiser depuis "CLIadmin" pour modifier la réservation dans "PlaceReservation".

7. Maintenant qu'on a un lieu et une date de cérémonie, le responsable communication va accéder à l'interface "PlanningManager" pour saisir la date de la cérémonie et une deadline pour toutes les inscriptions et absences (VIP, professeur, étudiant, accompagnateur et association).
8. afin d'informer les VIP et les professeurs, le responsable communication va accéder à "VIPInvitationManager", depuis le VIP, le component va accéder à Notifier qui rentre dans le component Notification. Idem pour informer les étudiants.
9. Le responsable communication va accéder à l'interface "InterludeManager" depuis "CLIadmin" pour créer les appels d'offre dans le component "Interlude".
10. L'association va accéder à l'interface "InterludeOfferer" depuis "CLIPublic" pour répondre à un appel d'offre dans le component "Interlude".
11. L'étudiant va accéder à l'interface "StudentAPI" depuis "CLISudent" pour accepter l'invitation dans le component "Student". A ce moment, l'étudiant peut aussi inscrire deux accompagnateur gratuitement, ou plus, mais en payant au service externe "Bank" à l'aide de l'interface "PaymentRequester".
12. Le VIP ou le professeur va contacter le responsable communication pour prévenir d'une absence ou d'un retard. Le service communication va accéder à l'interface VIPInvitationManager depuis "CLIadmin" pour prévenir de l'absence/présence du VIP/professeur dans le component "Interlude".
13. Le professeur va accéder à l'interface "PlanningViewer" depuis "CLIPublic" pour récupérer la liste de ses étudiants d'une filière dans le component "Planning".
14. La deadline est dépassé, ce qui veut dire que les étudiants ne peuvent plus s'inscrire et les associations ne peuvent plus répondre aux appels d'offre.
15. Le responsable communication va accéder à l'interface InterludeManager depuis "CLIAdmin" pour choisir une des associations qui a répondu à un type d'appel d'offre dans Interlude.
16. Le responsable communication va envoyer le nombre exacte de participant au centre des congrès, a quelque absence de professeur ou de VIP prêt. N'importe qui peut accéder au planning par à l'interface PlanningViewer depuis "CLIPublic".
17. Le responsable comptable va accéder à l'interface BillsManager depuis "CLIadmin" afin de gérer un justificatif de paiement dans le component "Bills".

F. Interfaces exposées

Composant : StudentManager

MinimalDateGenerator

void getStudents() // récupère les étudiants

Date obtainMinimalDate() // fournis la liste des étudiants au service d'impression et renvoie la date de livraison

Composant : Notification

Notifier

void notify(Student student) //Envoie les invitations à la cérémonie aux invités (Mail probablement)

void notify(Vip vip) //Envoie les invitations à la cérémonie aux invités (Mail probablement)

Composant : Student

StudentAPI

void invitePerson(Person person) //Sert à inviter une personne à la cérémonie

void acceptInvite()

PresentStudentGetter

List<Student> getPresentStudents() //Récupère la liste des étudiants venant à la cérémonie

InvitationTriggerer

void trigger() //déclenche la notification de tous les étudiants (invitation).

Composant : Planning

PlanningManager

void initiateNewCeremony(); // initialise une nouvelle cérémonie

void SetDeadLine(Date deadline) // pose la deadline et ouvre les inscriptions.

void SetDateCeremony(Date date) // pose la date de la cérémonie

PlanningViewer

Planning getPlanning() //Récupère le planning

Set<Student> getStudents() // renvoie la liste des étudiants qui participent à la cérémonie

Set<Student> getStudents(Speciality) // renvoie la liste des étudiants de la filière

Composant : VIP

VIPInvitationManager

void addVip(Vip vip) // ajoute un vip à la cérémonie courante

void addVip(Set<Vip> vips) // ajoute la totalité des vips.

void deleteVip(Vip vip) // supprime un vip

void triggerInvitation() // déclenche l'invitation de tous les vip

PresentVIPGetter

List<VIP> getPresentVIP() //Récupère les VIP n'ayant pas refusé l'invitation

Composant : Interlude

InterludeManager

void createInterlude(DateTime start, DateTime end);

Set<Club> getClubsForInterlude(Interlude) // récupère la liste des clubs pour un intermède.

void chooseClubOnInterlude(Club, Interlude) //Choisit le club qui participera à une interlude

InterludeOfferer

void offerInterlude(Interlude, Club) //Permet à une association de candidater sur un intermède

CompleteInterludeGetter

List<ClubInterlude> getInterludes() //Récupère les passages des associations qui ont été validées

Composant : PlaceReservation

ReservationVisualiser

int getPeopleQuantity() // retourne le nombre de personnes venant à la cérémonie (afin de prévenir le centre des congrès)

Composant : bills

BillsManager

void addBill(Bill bill) //Ajoute une facture dans le gestionnaire de factures

Set<Bill> getBill() // renvoie la liste des facture pour la cérémonie en cours.

external Composant : HyperPlanning

DiplomaChecker

List<Student> getDiplomedStudents() //Récupère la liste des étudiants ayant validé leur diplôme

external Composant : Printing Service

Printer

DateTime askPrint(Set<Student>) //Retourne la date à laquelle tous les diplômes des étudiants de la liste seront imprimés

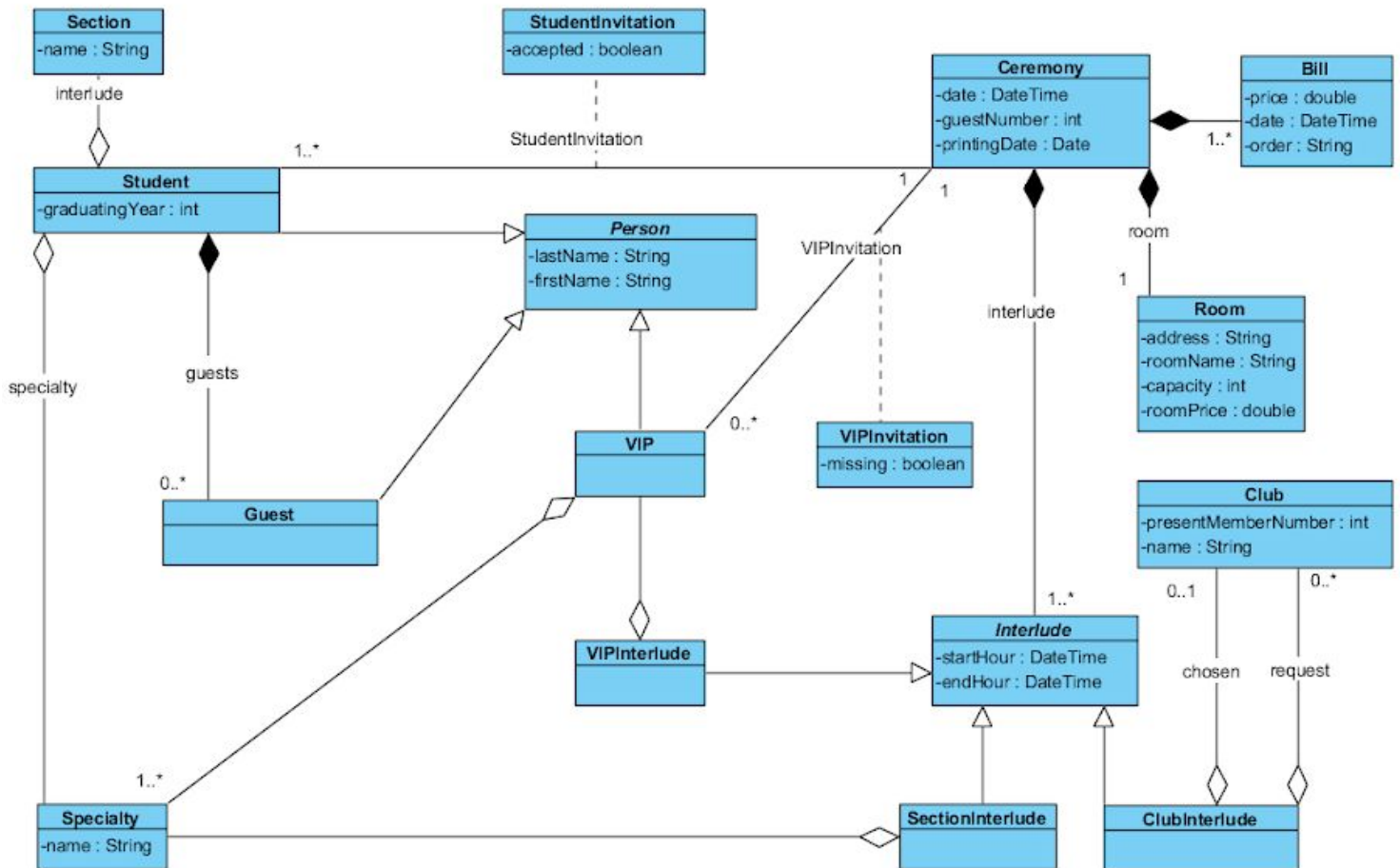
external Composant : Bank

PaymentRequester

String notifyPaymentRequested(double amount, Student student) //Demande un paiement à l'étudiant, renvoie le numéro de la transaction.

II. Vue développement

A. Diagramme de classe des objets métiers



Un "Guest", un "Student" et un "VIP" héritent de "Person" car ils possèdent des attributs en commun (lastName et firstName) et des méthodes spécifiques.

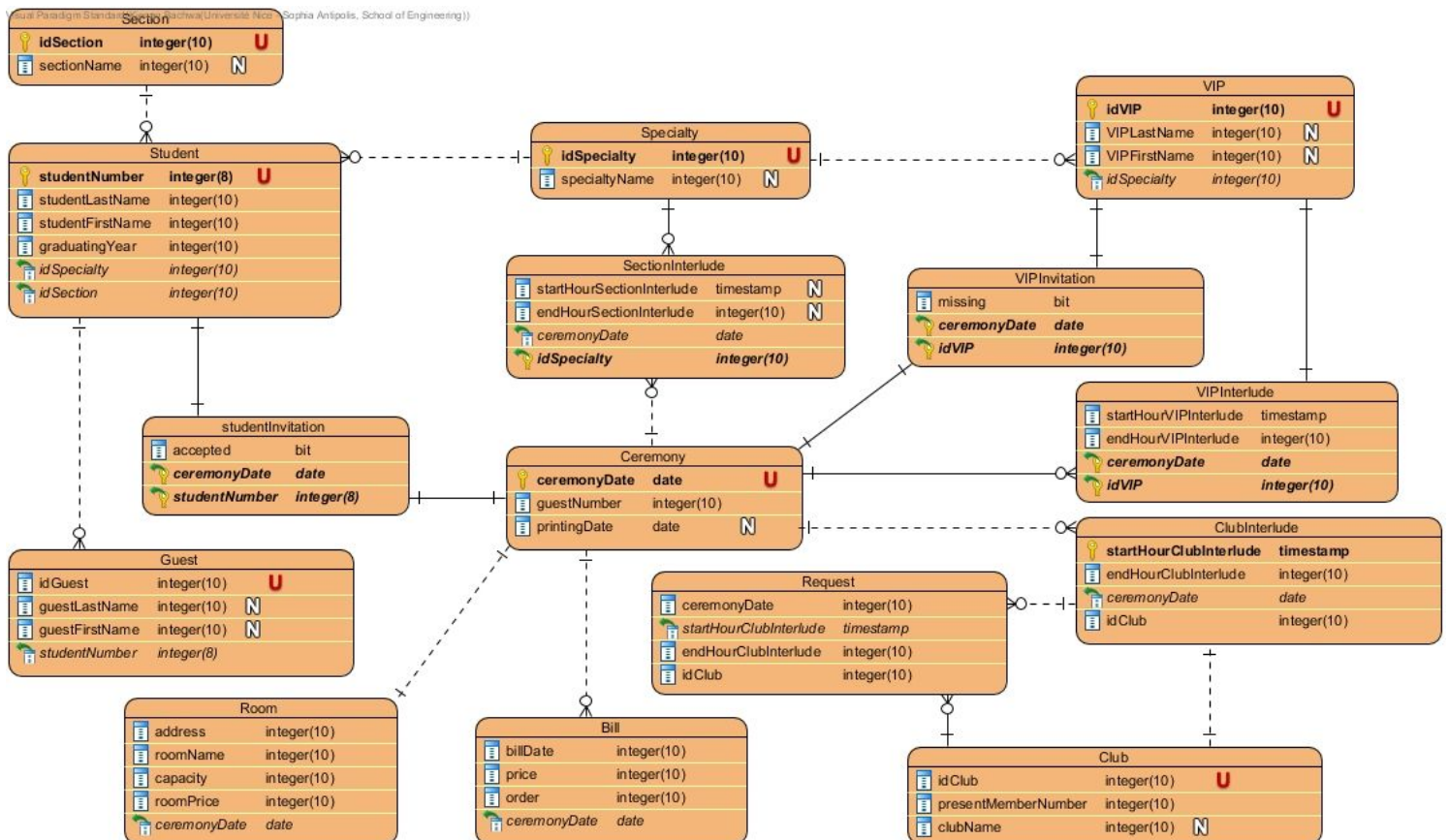
Les classes "VIPInterlude", "SectionInterlude" et "ClubInterlude" héritent de "Interlude" car ils possèdent des attributs en commun et sont chacun associé à une classe spécifique. De plus la méthode getDuration() définie dans "Interlude" ne se calcule pas de la même manière dans les différents classes filles.

La classe "Ceremony" utilisera les méthodes getDuration() de chacun des passages qu'il contient afin de calculer la durée totale.

La classe “StudentInvitation” est une classe associative entre “Ceremony” et “Student”. Elle permet de savoir si l’étudiant est absent pour mettre à jour le planning de la Cérémonie en conséquence.

De la même manière, un VIP et une cérémonie sont nécessaire pour créer une “VIPInvitation”. De plus, les invitations de VIP et des étudiants sont différentes car l’étudiant prévient de sa présence, tandis que le VIP prévient de son absence.

B. Modèle relationnel de stockage



C. Justification du mapping objet-relationnel

La table “Request” apparaît comme étant la collection d’associations (“Club”) ayant postulé pour “ClubInterlude”, c’est à dire une intermède. Une association peut postuler qu’une seule fois à une intermède, mais la table “Request” peut avoir plusieurs association pour une intermède.

Nous avons ajouté des id aux classes “Section”, “Specialty”, “Guest”, “Student”, “VIP” et “Club” car sans ces id, ces objets ne pourront pas être unique. Par exemple, dans la classe “Guest”, un étudiant peut avoir deux personnes avec lui ayant le même prénom et le nom.

A propos de la classe “Ceremony”, la date suffit en tant qu’identifiant unique car il n’y aura pas plus d’une cérémonie par jour (et voire pas plus d’une cérémonie par an).

Les clés primaires des classes d'association "StudentInvitation" et "InvitationVIP" sont composées des deux clés primaires qui les associent.

Les classes "VIPInterlude", "SectionInterlude", "ClubInterlude" sont le lien entre la cérémonie et une autre classe (respectivement "VIP", "Section", "Club"). Leur clé primaire sera donc composée de la clé primaire de "Ceremony" et de celle de l'autre classe.

Pour la classe "Room", la clé primaire est composée de l'adresse et du nom de la salle, car pour une adresse donnée les salles auront des noms différents. Par conséquent, il n'y a pas besoin d'ajouter un id.

Pour la classe "Bill", les factures auront une date/heure unique et seront associées à une "Ceremony", donc la clé primaire est composée de la clé primaire de la "Ceremony" associée et de la date de la facture. De plus, les timestamp permettent de stocker l'heure en plus du jour comparé à date qui ne stocke que le jour.

Conclusion

Le travail effectué pour ce rapport nous a permis de réfléchir à l'organisation de notre code et des différentes classes que nous utiliserons. Cependant ces diagrammes sont voués à changer car nous allons sûrement trouver des contraintes techniques et les besoins du client pourraient évoluer.