



# J-CONFS

FLORYAN KIEFFER, MAJID ACHRIA, SEBASTIEN BOURG, ANIS HAMOUNI, ANIS ZOUARI

2019-2020

# SOMMAIRE

1. Projet J-Confs
  - a. Présentation du projet
  - b. L'existant
2. Evolutions apportées
  - a. Récupération informations des chercheurs
  - b. Extension du package calendar et conference
  - c. Ajout du package location
3. Démonstration du code
4. Difficultés rencontrées

# 1. PROJET J-CONFS

## a. Présentation du projet

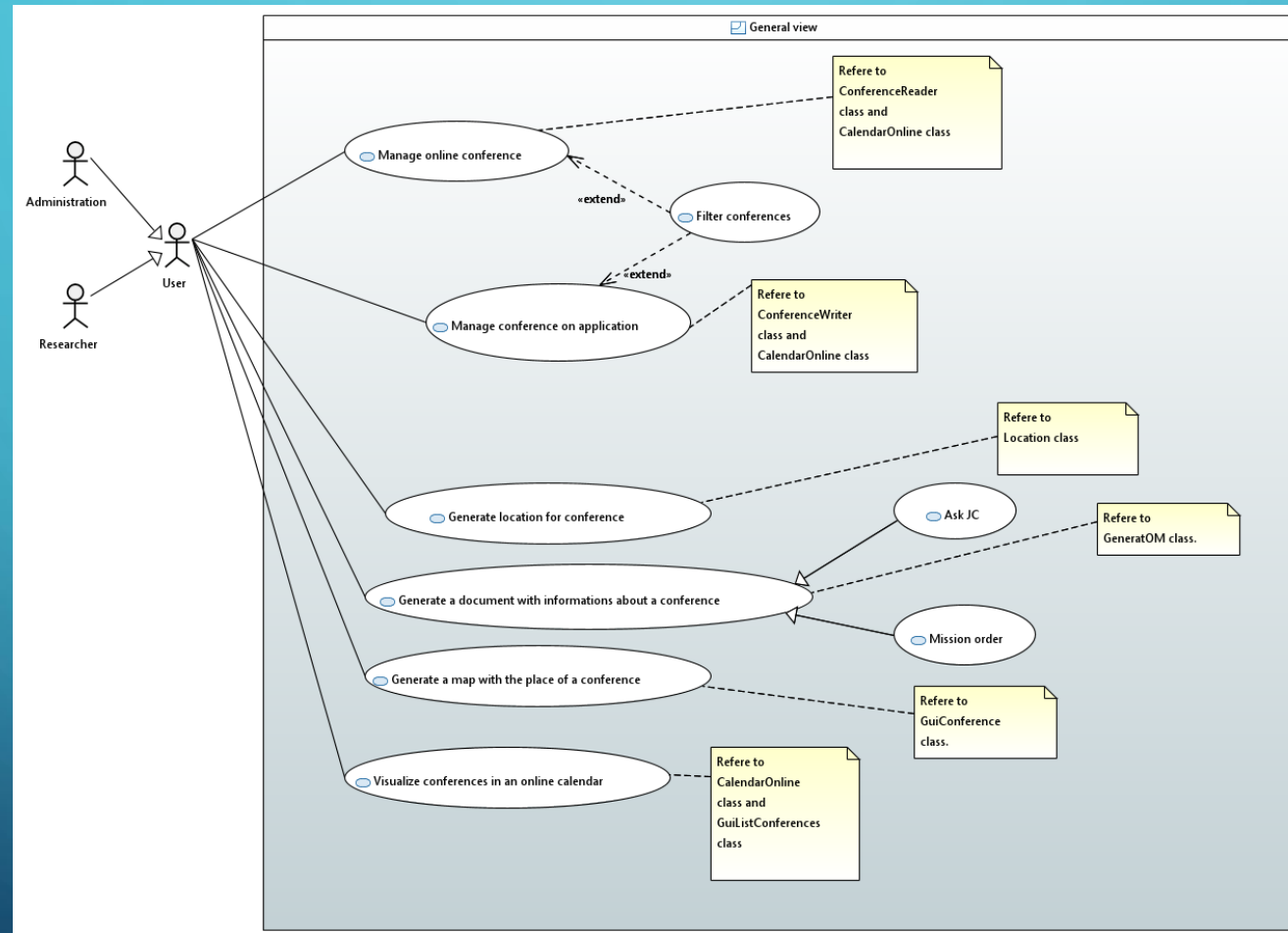
- Outil de gestion de déplacement en conférence
- Départ d'un projet existant
- Utilisation dans un premier temps en interne (Dauphine)
- Utilisé par les chercheurs et l'administration

# 1. PROJET J-CONFS

## b. L'existant

- Stocker et lire des conférences dans frux
- Affichage partiel du lieu d'une conférence sur une carte
- Génération de document sur base des coordonnées d'un chercheur
  - Non fonctionnel
- IHM

# DIAGRAMME DE CAS D'UTILISATION GÉNÉRAL



## 2. EVOLUTIONS APPORTÉES

### a. Récupération informations des chercheurs

- Impossibilité de récupérer des données sur un chercheur
- Communication avec le responsable du pôle infrastructure et intégration
- Correction de l'erreur en modifiant le lien

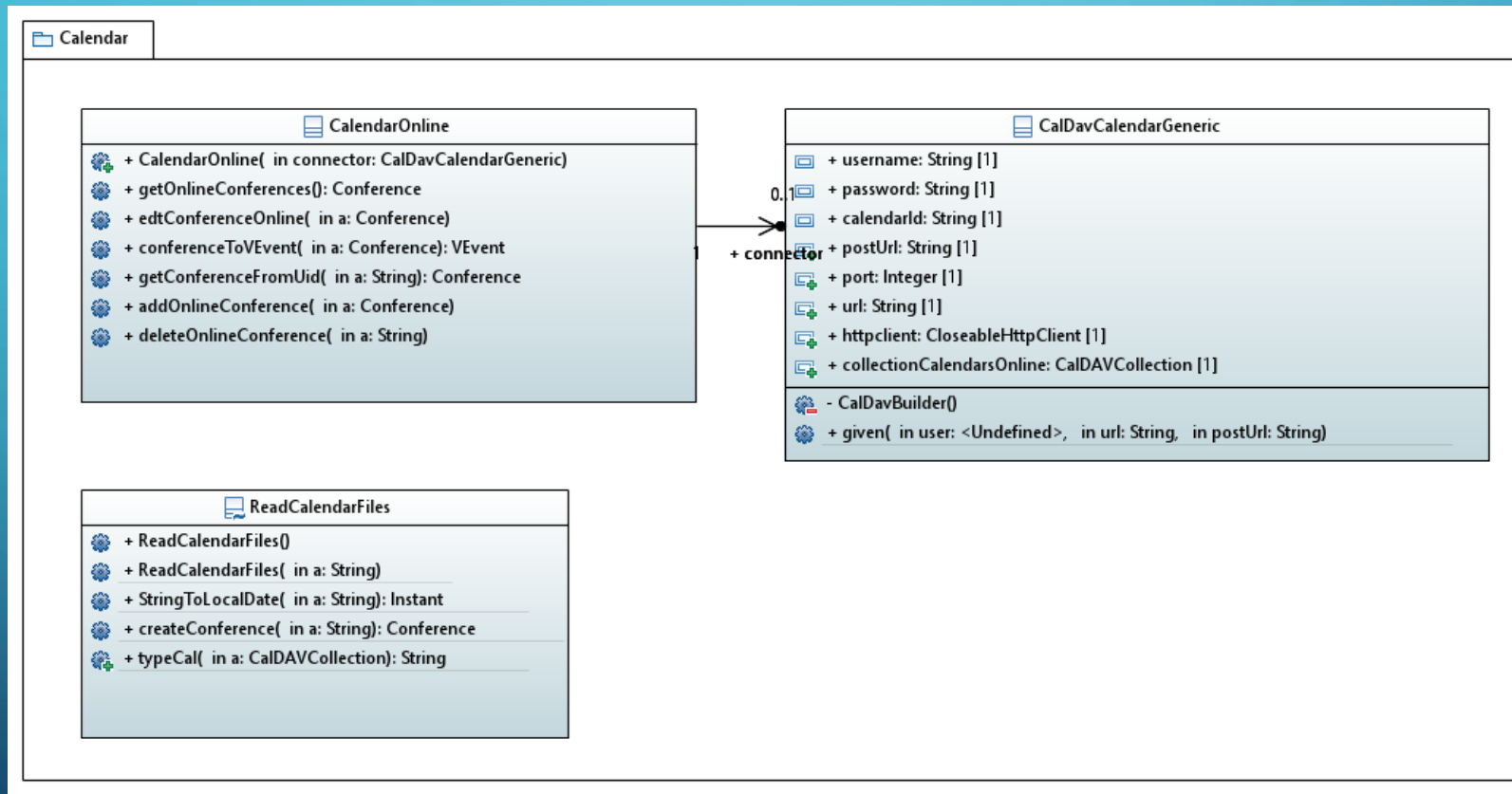
## 2. EVOLUTIONS APPORTÉES

### b. Extension du package calendar et conference



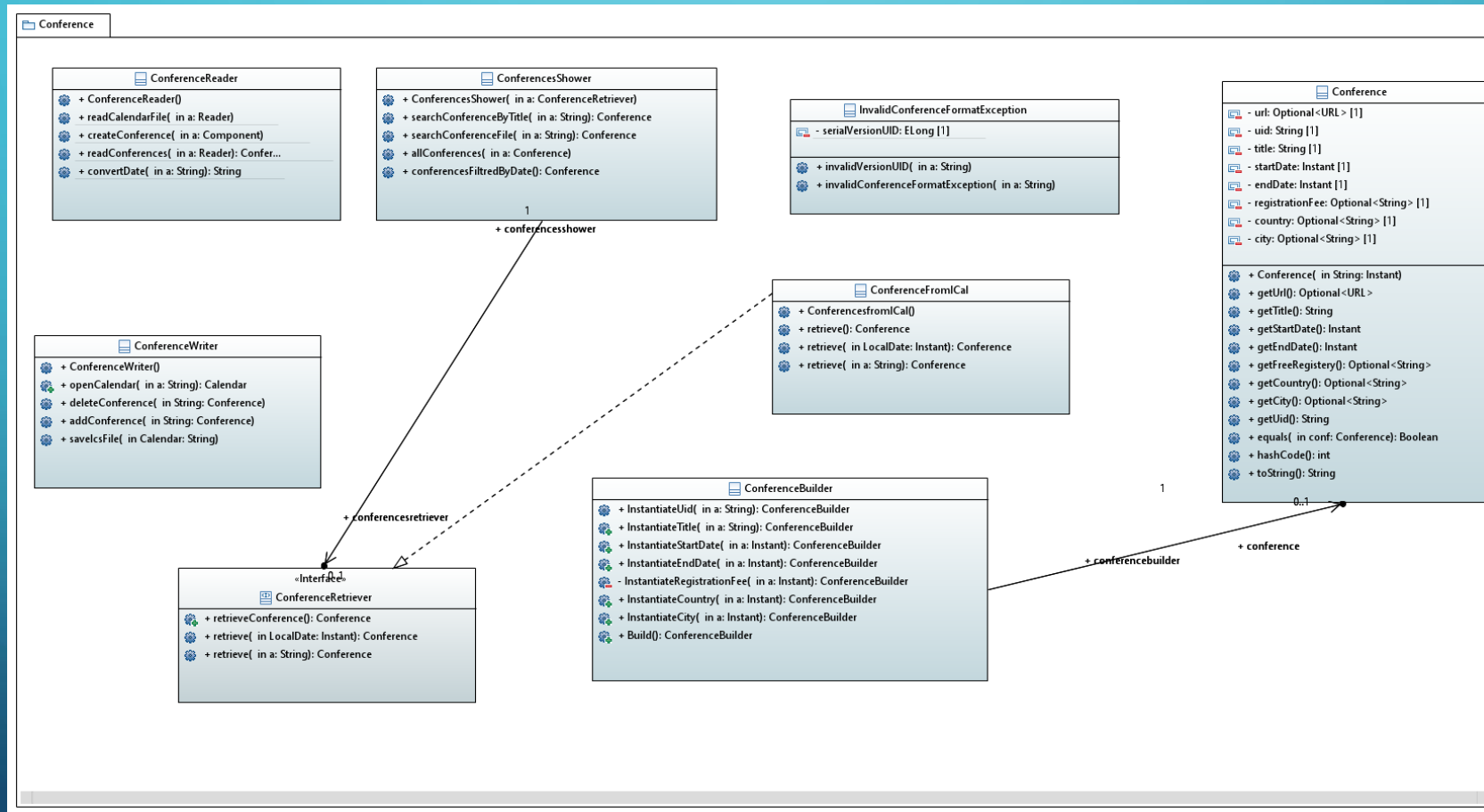
- Uniquement fruux au début
- Rechercher des hébergeurs de calendrier en ligne
- Compréhension de connexion entre l'application et les serveurs de calendrier
- Mise en place du code
- Possibilité de se connecter sur fruux et tous les hébergeurs de la plateforme NextCloud

# DIAGRAMME DE CLASSE CALENDAR





# DIAGRAMME DE CLASSE CONFERENCE

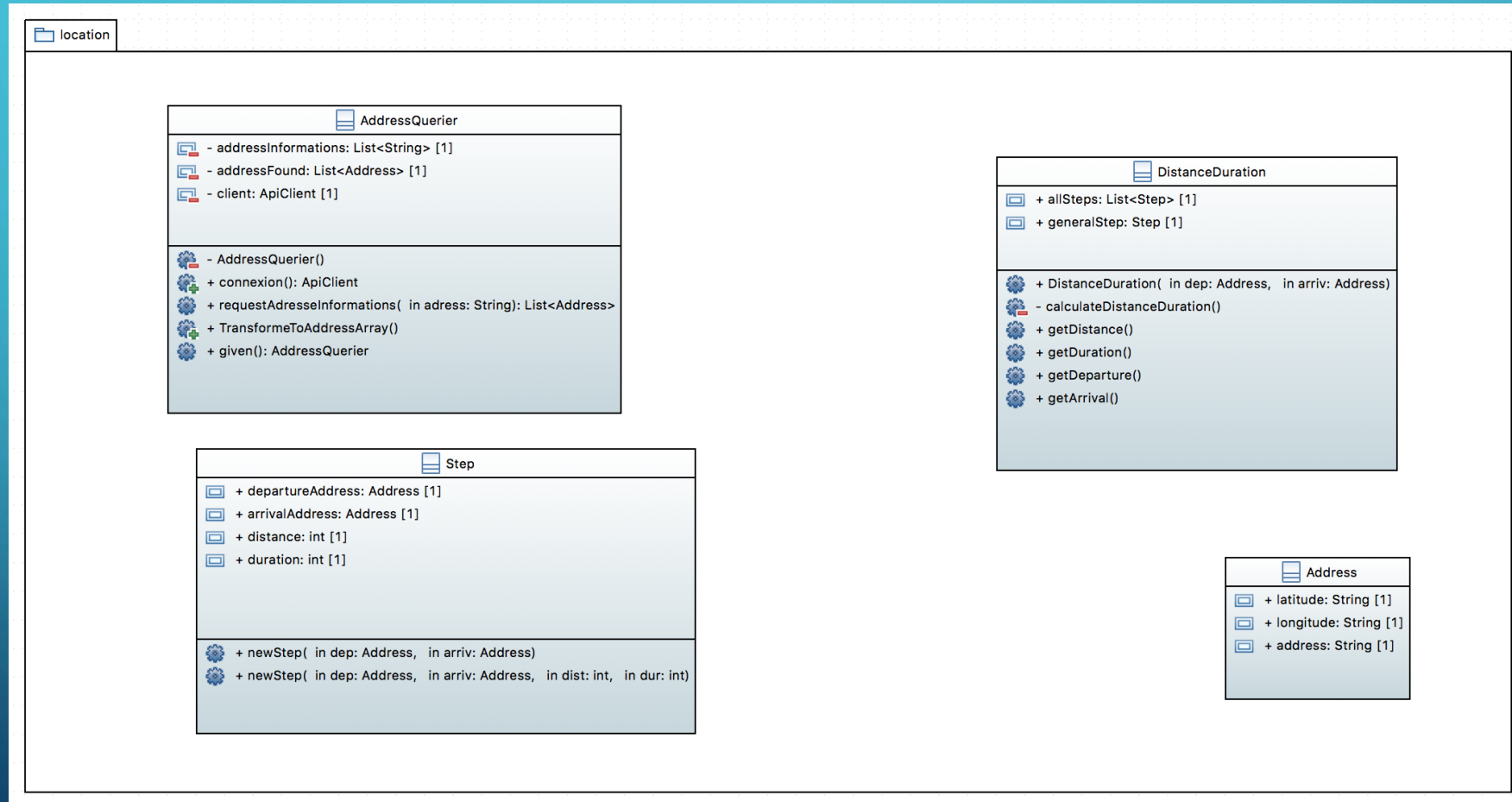


## 2. EVOLUTIONS APPORTÉES

### c. Ajout du package location

- Remplace le package geocode
- Importation manuelle de l'API LocationIQ car pas de dépendance Maven
- Objectif : Mettre en place un itinéraire entre deux adresses
- 4 classes: AddressQuerier, Address, Step, DistanceDuration

# DIAGRAMME DE CLASSE LOCATION



### 3. DÉMONSTRATION DU CODE

- Nous allons vous présenter les fonctionnalités ajoutées via les tests unitaires avec le résultat qui s'affichera dans la console ou l'interface web du calendrier.



## 4. DIFFICULTÉS RENCONTRÉES

### JAVA

- Import de locationIQ
- Peu de documentation des API et du code existant
- Prise en compte de toutes la documentation (application BP)

### UML

- Papyrus nouveau langage
  - Diagramme de séquence
  - Pas intuitif
- Représenter correctement les diagrammes de cas d'utilisation
- Lier les différents diagrammes

## 4. DIFFICULTÉS RENCONTRÉES

### PROJET

- Gérer en équipe un projet universitaire du début à la fin
- Se rendre compte de la complexité
- Nécessité que le code soit complétement conforme aux conventions
- Respecter les délais de livraison
- Difficulté de compréhension entre l'équipe et le client

# CONCLUSION

- Projet intéressant : montée en compétences en Java, UML et gestion de projet
- Pas de GUI réalisé mais solidification du code orienté API
- Nombreuses opportunités intéressantes pour le prochain groupe