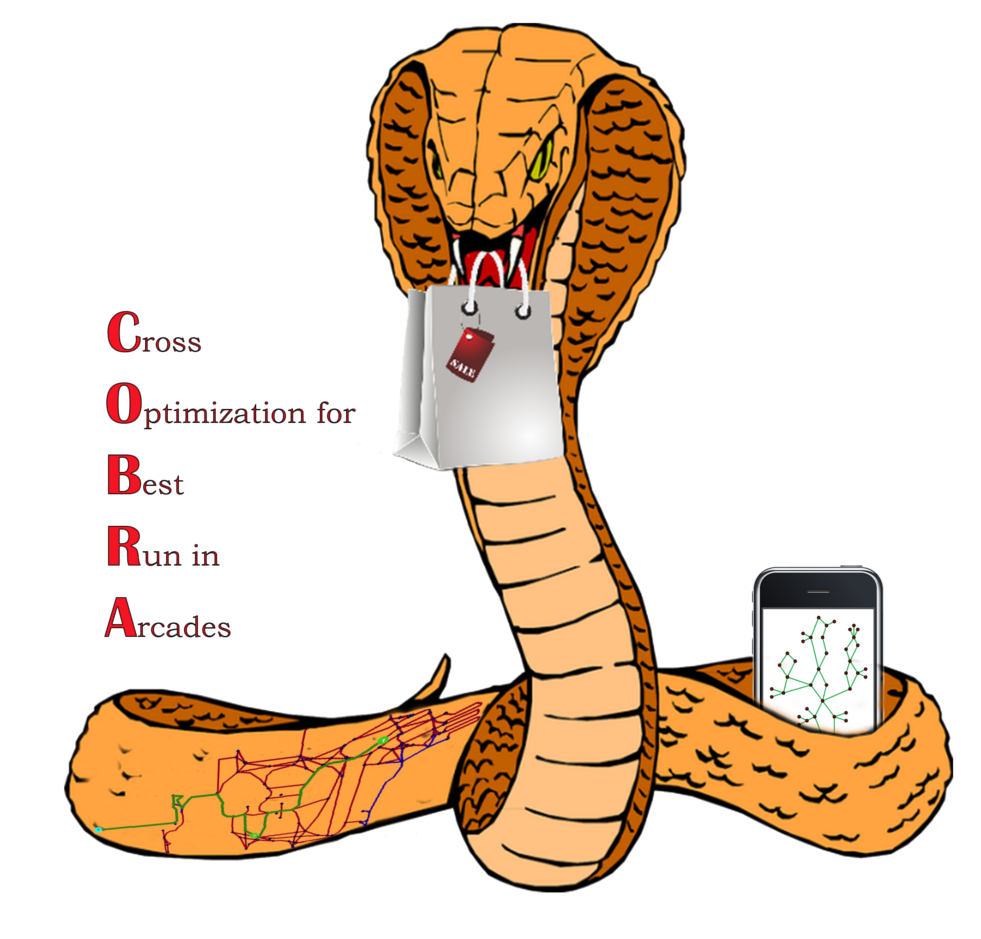
**MANUEL DE DEVELLOPEMENT DE L’APPLICATION C.O.B.R.A.**



Sommaire :

[1. Modélisation : 3](#_Toc450902250)

[1.1 Cas d’utilisation : 3](#_Toc450902251)

[1.2 Diagramme de classes : 4](#_Toc450902252)

[1.3 Diagramme d’activité : 5](#_Toc450902253)

[1.4 Diagramme de séquence : 6](#_Toc450902254)

[2. Organisation et fonctionnement du code : 7](#_Toc450902255)

[2.1 Organisation général : 7](#_Toc450902256)

[2.2 Fonctionnement des différentes parties : 7](#_Toc450902257)

[2.2.1 MainActivity.java : 7](#_Toc450902258)

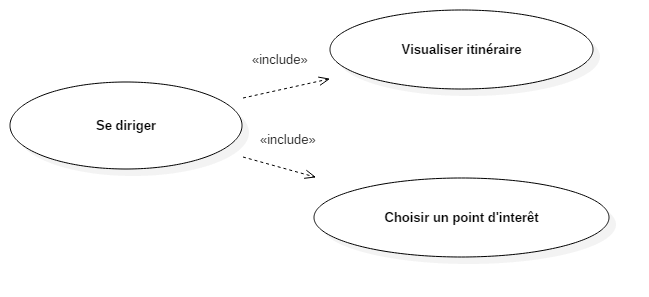
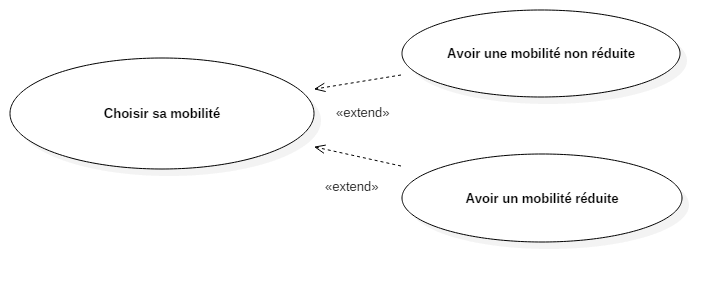
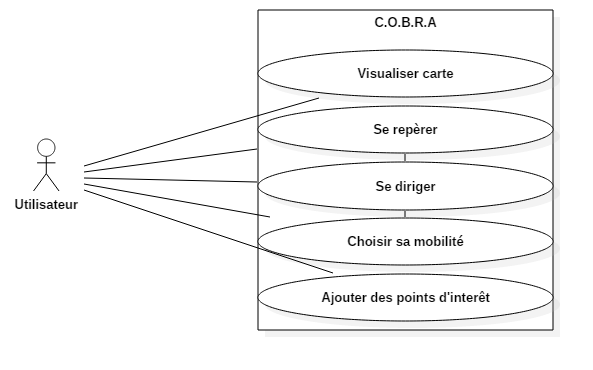
# Modélisation :

## Cas d’utilisation :

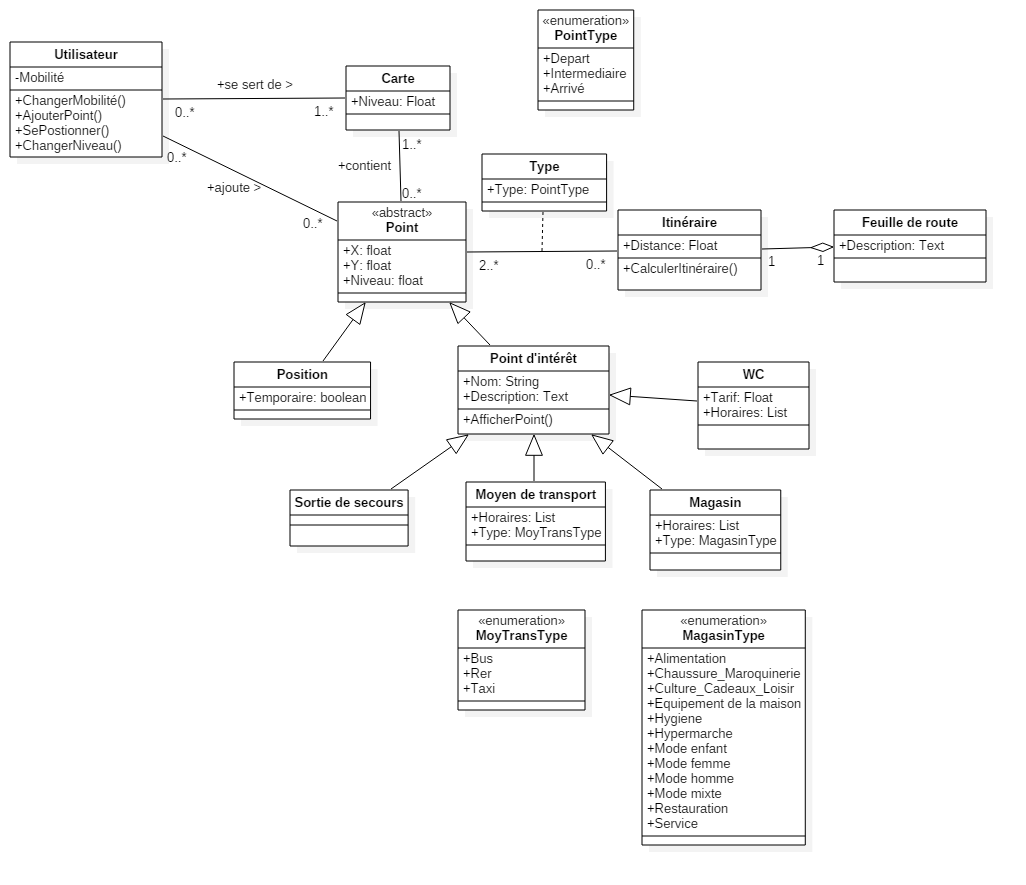
Notre application doit aider un utilisateur à effectuer différentes tâches dont les principales sont listées ci-dessous :

* Visualiser la carte du centre commercial
* Se repérer dans ce centre
* Se diriger à l’intérieur des Arcades
* Choisir sa mobilité, c’est-à-dire si il a une mobilité réduite (handicap, poussette, caddie) ou non
* Ajouter des points d’intérêt

Vous trouverez ci-dessous les diagrammes de cas d’utilisation représentant les fonctionnalités principales de notre application COBRA (Cross Orientation for Best Run in Arcades) :



## Diagramme de classes :



Nous avons identifiés 6 classes principales :

* Utilisateur
* Carte
* Point (Dont position et point d’intérêt est une classe enfant)
* Point d’intérêt (Dont WC, sortie de secours, moyen de transport, magasin sont des classes enfants)
* Itinéraire
* Feuille de route

Nous sommes partis de données préexistantes notre modélisation s’est donc basée sur ses données, notamment pour les classes Point et Point d’intérêt.

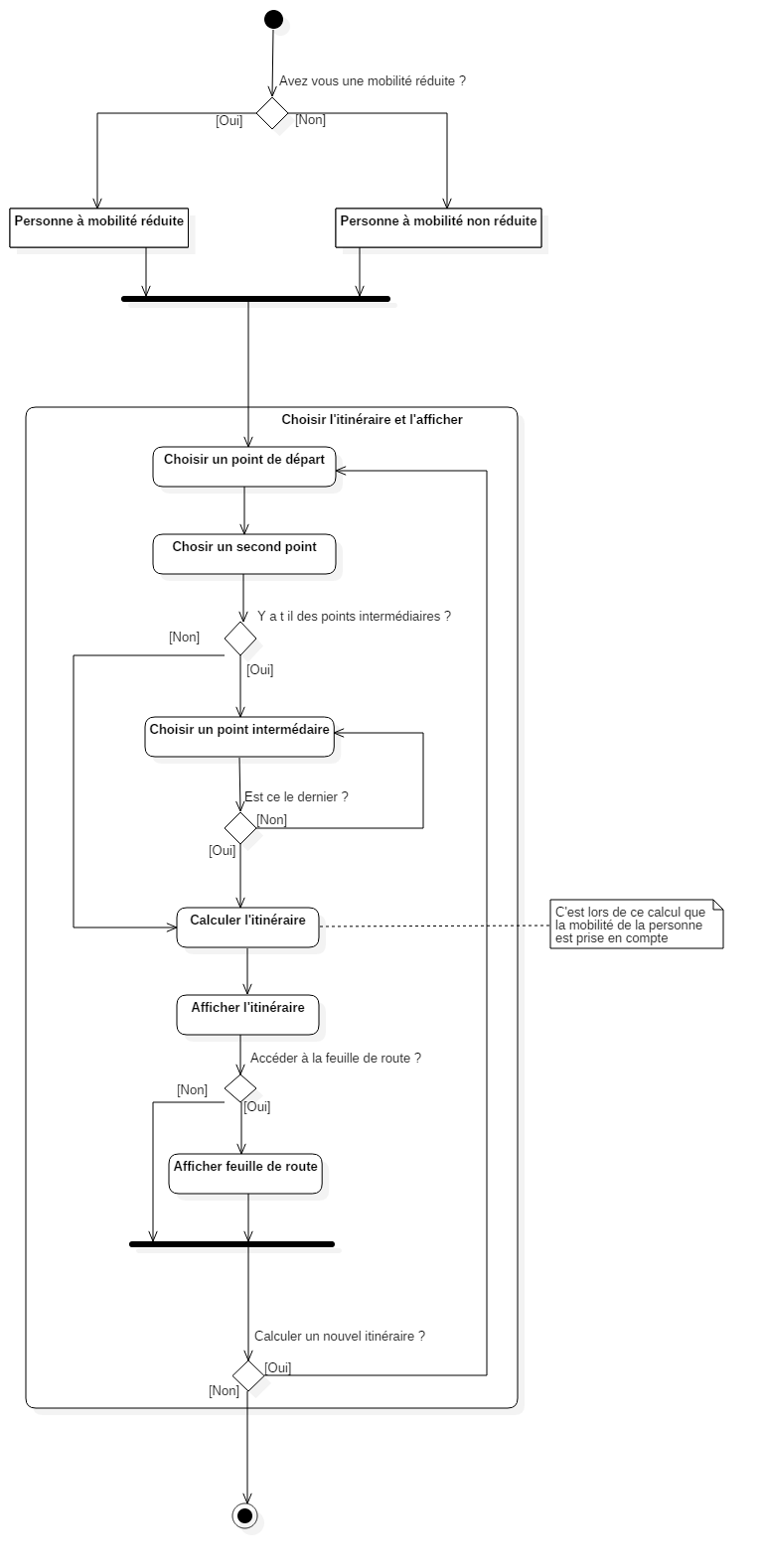
Les classes Itinéraires et Feuille de routes servent à afficher les informations nécessaires à l’Utilisateur pour se déplacer. (Feuille de route n’étant pour l’instant pas utilisée dans notre application).

La classe Carte sert à la gestion du multi-niveau, en effet le but de notre projet est de gérer un itinéraire sur du multi-niveau.

Enfin la classe Utilisateur sert à gérer la mobilité de ce dernier (s’il peut ou non monter des escaliers).

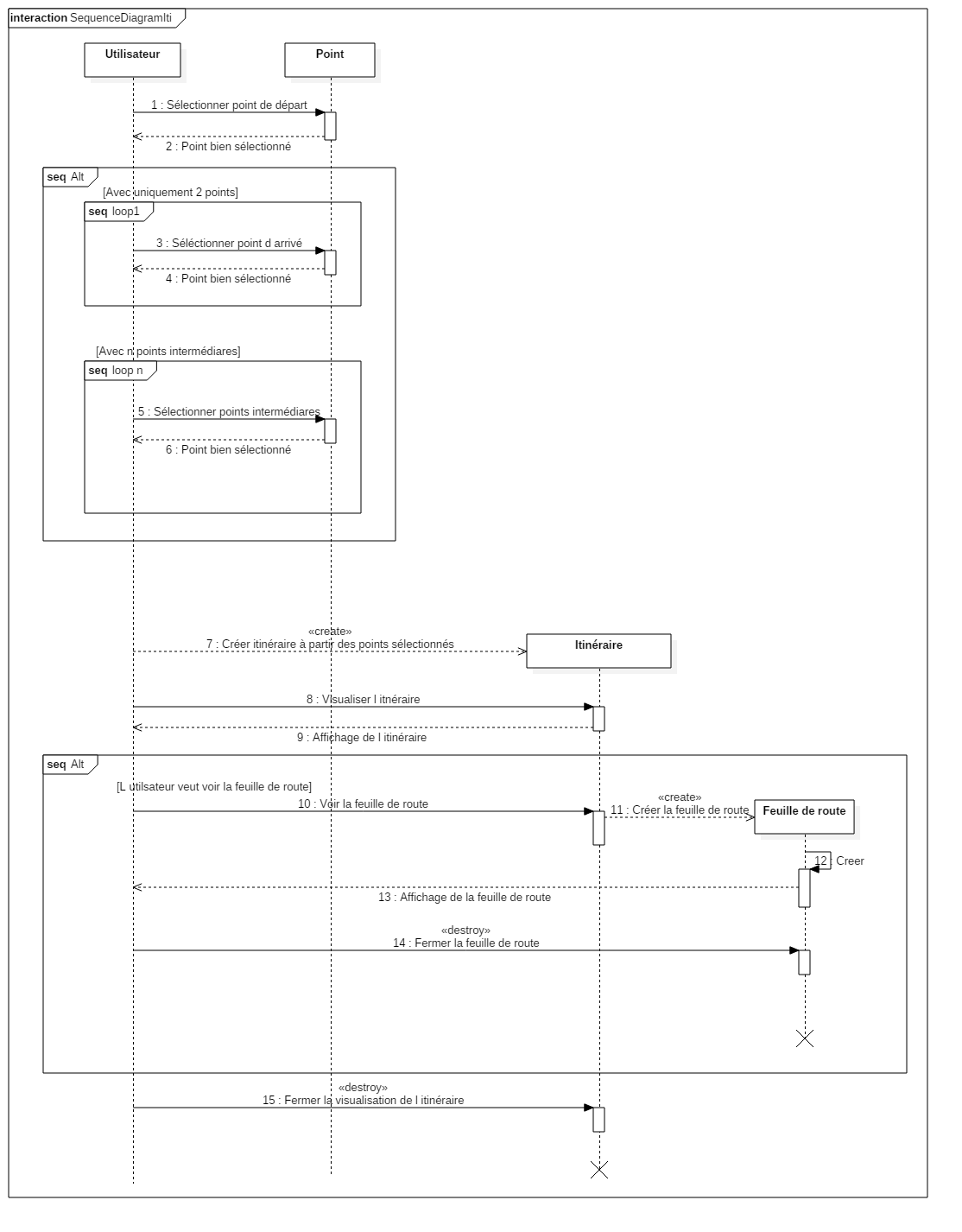
## Diagramme d’activité :

Ce diagramme présente le déroulement de la création d’un itinéraire sur notre application



Notez cependant que le choix de point intermédiaire et l’affichage de feuille de route n’est pas gérer dans la version actuelle de notre code.

## Diagramme de séquence :



Dans notre application, nous commençons par choisir un point de départ, puis nous choisissons le point d’arriver, (le choix de point intermédiaire n’étant pas encore implémenter). A ce moment le calcul d’itinéraire est effectué. Puis la feuille de route est créée et affichée (Non implémentée dans la version actuelle de notre application).

# Organisation et fonctionnement du code :

## Organisation général :

Nous allons vous présenter ici l’organisation général du code, le fonctionnement de chacune des parties décrites sera développée par la suite.

* Dans AndroidProjet\app\src\main\java\com\example\formation\androidprojet\_v1 nous avons les fichiers java qui gèrent le fonctionnement de notre application:
  + MainActivity.java : gère la majeure partie fonctionnement de notre application (récupération de la base de données, calcul et affichage d’itinéraire par niveaux, formulaire avec autocomplétion).
  + Choix.java, Listetype.java, Listemagasin.java : servent à gérer le formulaire par choix du type de magasin (et le point de départ par QrCode).
  + Nous lançons d’abord Choix.java où nous avons le choix (héhé le nom est bien choisi) entre trouver un magasin par type pour le départ et l’arrivé, ou par scan de QrCode pour le point de départ. Le choix par type lance Listetype.java qui affiche la liste des types de magasins, lorsque l’on choisit un type on lance Listemagasin.java qui affiche la liste des magasins de type le type sélectionné. Lorsque qu’un magasin est sélectionné, on détruit l’activité lié à Listemagain.java et on envoie le résultat à l’activité de Listetype.java et de même on renvoie le résultat à l’activité de choix.java. Lorsque l’on clique sur le bouton, on est renvoyé sur MainActivity.java, où si deux points ont été sélectionnés alors l’itinéraire et calculé et affiché.
* Dans AndroidProjet\app\src\main\res\layout nous avons les fichiers liées à l’affichage des différentes activités, les différents fichiers correspondent aux différents fichiers java décrits ci-dessus.

## Fonctionnement des différentes parties :

### 2.2.1 MainActivity.java :