

PROJET GL2
GM-MI SEMESTRE 2
Application COVID-19

Zaouche Djaouida

May 12, 2020

0.1 Description du contexte du projet

Depuis plusieurs semaines, nous vivons une crise sanitaire sans précédent qui a fortement impacté notre quotidien et nos habitudes. Dans ce contexte, de nombreuses données sont collectées et peuvent être trouvées sur le site : <https://www.data.gouv.fr>. À titre d'exemples, on peut trouver des données par :

- <https://www.data.gouv.fr/fr/reuses/covid-19-par-regions/>,
- <https://www.data.gouv.fr/fr/datasets/donnees-des-urgences-hospitalieres-et-de-sos-medecins-relatives-a-lepidemie-de-covid-19/>,
- <https://www.data.gouv.fr/fr/datasets/coronavirus-covid19-evolution-par-pays-et-dans-le-monde-maj-quotidienne/>.

L'application à réaliser sera utilisée quand des mesures radicales sont prises où certaines communes imposent le confinement à ses habitants et restreignent drastiquement la circulation automobile. Pour une commune confinée, les habitants ont l'obligation de rester chez eux. De plus, les transports publics sont suspendus ainsi que les autocars et trains longue distance. En d'autres termes, les grands axes routiers sont entièrement fermés à la circulation.

L'objectif de ce projet est de concevoir et développer une application d'aide aux personnes habitant des communes non confinées désirant se rendre à d'autres communes non confinées.

Votre application devrait aussi donner des informations graphiques qui seront détaillées plus loin.

0.2 Vers une bonne conception

Une première ébauche d'un diagramme de classes est illustrée par la Figure 1. Il vous est demandé de compléter ce diagramme. Dans ce qui suit, nous donnons une description générale des méthodes à implémenter :

- ***comConf()***¹ (*): donne la liste des communes confinées. La liste doit

¹* simple, ** difficulté moyenne, *** plus d'efforts.

être triée par durées de déconfinement prévisionnelles restantes dans l'ordre croissant.

- ***comNonConf()*** (*): donne la liste des communes non confinées triée par dernières dates de déconfinement dans l'ordre décroissant. Observons qu'une date de déconfinement peut être vide.
- ***confiner(date, commune)*** (*) : permet de rendre une commune confinée.
- ***deconfiner(date, commune)*** (*) : permet de rendre une commune non confinée.
- ***carteColoreTaux(date)*** (**): affiche une carte visualisant les communes (utilisez les API google map). Chaque commune est colorée en rouge, vert ou orange selon l'étendue de l'épidémie dans la commune. Vous pouvez enrichir le diagramme de classes afin de disposer d'informations pertinentes pour définir des règles de coloration.
- Définir d'autres méthodes calculant des paramètres utiles pour la prise de décision (confinement, déconfinement) tels que les taux de guérison pour une période donnée. Pensez à exploiter les objets de la classe "Historique". Il est évident qu'un objet "Historique" concerne les données d'une commune pour un jour donné.
- ***statistique()*** (**): affiche des histogrammes pour donner des informations sur les communes, en termes de confinement et déconfinement, des différentes régions du pays, par exemple, des histogrammes pour visualiser :
 - les taux de communes confinées,
 - les taux des communes jamais confinées ².
- ***chemin(ComNonConf, ComNonConf): ArrayList<ComNonConf>*** (***) : donne le plus court chemin entre deux communes non confinées. On suppose qu'il est strictement interdit de traverser une commune

²Chaque histogramme est dédié à une date donnée. Chaque barre représente un taux calculé pour une région donnée. De plus, la date doit apparaître dans le titre de chaque histogramme.

confinée et que les chemins entre les communes ne sont pas orientés. Ainsi un tel chemin peut être vide. Observez que les chemins modélisés dans le diagramme de classes ne sont pas valués. Pensez aux classes d'association pour le faire.

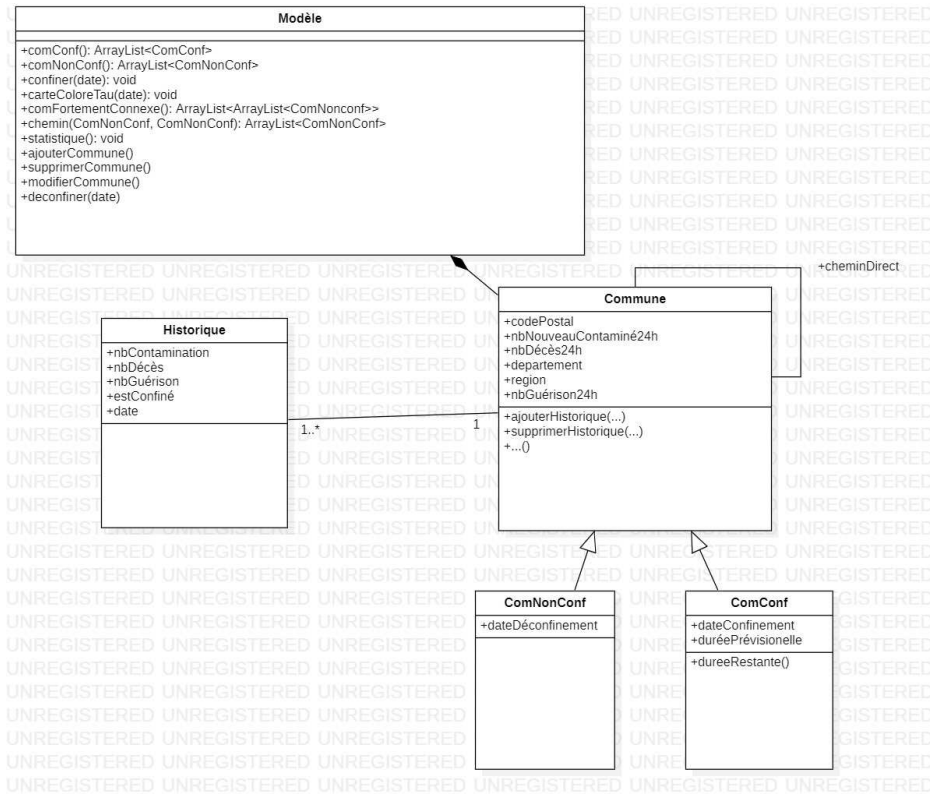


Figure 1: Un noyau du diagramme de classe

Les méthodes `toString`, `hashCode` et `equals` doivent être redéfinies. Pensez à justifier le choix des tables (`ArrayList`, `hashSet`, `hashMap`, ...). Toutes les méthodes de la classe `Modèle` seront testées pour un objet singleton, appelé *france*. Pour les interfaces de votre application, seules deux interfaces graphiques sont exigées (l'interface d'accueil et une seconde).

0.3 Une modeste contribution vers la prédiction

Il vous est demandé de trouver les bons datasets et d'apporter des réponses aux questions suivantes :

- Prédire les survivants en utilisant les variables âge, sexe, problèmes de santé, région.
- Étudier l'importance des variables : quelles variables influent le plus sur le résultat du point précédent?
- Étudier l'évolution du nombre de malades dans une région donnée.
- Toutes les bonnes idées sont les bienvenues.

Nous vous recommandons le site <https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset/tasks>). Nous nous intéressons qu'aux régions de France. Vous pouvez suivre les étapes suivantes :

1. Collecte de données : récupérez des données concernant les différents facteurs intervenant dans l'évolution de la maladie vers la guérison ou le décès.
2. Statistique descriptive : appliquez divers outils de statistiques descriptives pour résumer vos données. Ces résumés seront présentés de manière graphique (histogramme, camembert, ...)
3. Data mining/Machine learning : appliquez une méthode de votre choix (Arbre de décision, Classifieur bayésien, ...) pour prédire les survivants. Analysez les résultats et étudiez (par une méthode que vous expliquerez) l'importance des différents facteurs dans l'issue de la maladie.

Vous pouvez vous inspirer de l'exemple de Titanic étudié en cours de Data mining.

0.4 Livrables

Le livrable comprend un rapport écrit, le code source de l'application :

- le diagramme de classes UML,
- le Fichier jar,
- les fichiers java,
- rapport technique,
- rapport de synthèse du groupe et de chaque membre.

Recommendations

- Veuillez à bien structurer votre code :
 - respecter le modèle *MVC*;
 - bien commenter les parties pertinentes de votre code;
 - et appliquer une bonne indentation.
- Veuillez à bien décrire votre application :
 - bien tracer les fonctionnalités de l'application;
 - bien resenser les acteurs ou utilisateurs de votre système et le rôle de chacun;
 - définir les contraintes d'intégrité des données et les règles de gestion à respecter;
 - faire des captures d'écran des différentes vues de votre application avec des descriptions compactes et cohérentes.

0.5 Conclusion

- On ne devient pas champion sans suer (Épictète).
- Le travail individuel permet de gagner un match mais c'est l'esprit d'équipe et l'intelligence collective qui permet de gagner la coupe du monde (Aimé Jacquet, ancien joueur et entraîneur international français de football)
- On n'est pas plus grand parce qu'on triche (Louis-Philippe Robidoux).