Checkpoint 1

Projet Transverse I

Gogniat Aaron

Table des matières

Introduction au projet

À la suite du cours de Design Science du Professeur Jean-Henry Morin que j'ai suivi le semestre passé (Automne 2023), j'ai rendu un projet de recherche intitulé « Ride-Quest : a proposal for a mobile application to push people to travel using a ride sharing and public transportation system ». Le but de ce projet a été de penser une application mobile qui permettrait à une personne d'atteindre sa destination à l'aide de co-voiturage et des transports publics en une course, dans le but d'inciter les personnes a utilisé plus régulièrement les transports en commun. Les passagers n'auraient pas à payer pour ce service mis a part leur billet de transport en commun, et les conducteurs se verraient récompenser de diverses manières. J'ai donc rendu une proposition d'un système qui permettrait en partie l'implémentation d'une telle application. J'ai dans l'idée de continuer cette proposition pendant ce cours de Projet Transverse 1 en sélectionnant le projet n°10 (« Système de réservation de taxi ») car celui-ci permet un accomplissement partiel de mon objectif final, qui est de proposer une application entièrement implémentée et fonctionnelle pour mon projet de Bachelor.

Description du projet et de sa problématique

L'objectif du projet est donc de créer un système qui matcherait un conducteur au passager qui a fait une demande pour une course. Le conducteur doit être choisi parmi d'autres conducteurs ayant acceptés la course, doit pouvoir recevoir les informations nécessaire pour conduire le passager à une plateforme de transport public qui pourrait ensuite l'amener à la destination définie, de le rejoindre à la station convenue pour ensuite le conduire à la destination renseignée, ou de le conduire de l'origine à la destination si aucune option de transport en commun ne peut être trouvé ou si le passager a choisi de ne pas utiliser ce type de transport.

En ce qui concerne ce projet, il va falloir designer et implémenter un système de réservation de covoiturage. Pour cela, il sera nécessaire d'implémenter un API qui permettra le choix d'un itinéraire ainsi qu'un algorithme de triage selon des critères de sélection qui sera utilisé pour trouver le conducteur correspondant le plus au passager. Il faudra également récolter les données GPS des utilisateurs en tout temps pour permettre de les afficher sur une carte ainsi que de les transmettre à leur pair pour permettre la bonne réalisation de la course.

Analyse des objectifs (besoins fonctionnels/non fonctionnels)

Modélisation des objectifs

Le modèle des objectifs permet d'identifier, explorer, évaluer et opérationnaliser les objectifs organisationnels et stratégiques [1].

Nous pouvons définir trois objectifs principaux : permettre la réservation d'une course (Objectif 1), réunir un conducteur et un passager (Objectif 2) ainsi qu'organiser un itinéraire (Objectif 3). Même si ces deux derniers objectifs sont visualisés comme des sous-objectifs, ceux-là n'en restent pas du moins très importants. Ces trois objectifs formant la base de ce sytème, nous pouvons ensuite décrire les sous-objectifs. Les sous-objectifs pertinents à mentionner sont « Permettre au passager de placer une demande de course » (Objectif 18), « Permettre à

l'utilisateur de choisir entre être conducteur et être passager » (Objectif 15), « Implémenter un algorithme d'itinéraire » (Objectif 8), « Suivre en temps réels les coordonnées GPS des utilisateurs » (Objectif 13), « Implémenter un algorithme de matching » (Objectif 9) et « créer une base de données » (Objectif 12).

En ce qui concerne l'implémentation d'un algorithme d'itinéraire (Objectif 8), trois opportunités ont été soulevées. En effet, les APIs mis à notre disposition par les différents services de transport en commun ainsi que Google Maps (Opportunité 1-3) permettront de trouver l'itinéraire le plus cohérent aux différentes informations récoltées.

Trois obligations ont été relevées. Il s'agit d'obligations légales concernant l'âge et les papiers légaux des utilisateurs. Ainsi, le conducteur doit être en âge de conduire (Obligation 1), être muni d'un permis de conduite valable (Obligation 2) et le passager doit avoir âge légal pour utiliser un tel service (Obligation 3).

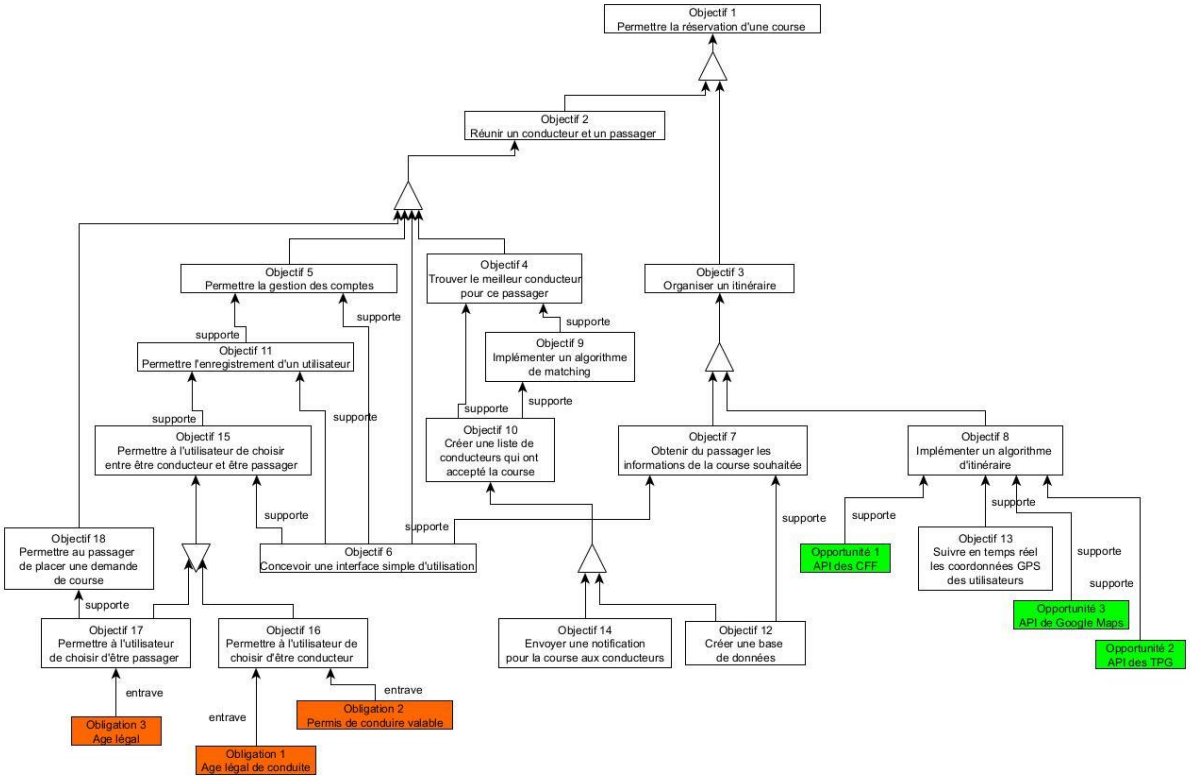

Figure 1 : Modèle des Objectifs

Liste des acteurs

La liste des acteurs nous permet d'identifier les différents membres du système.

Nous n'avons qu'un type général d'acteur, c'est-à-dire l'utilisateur de notre système. Celui-ci peut être spécialisé en conducteur, qui offre une course, et passager, qui demande une course.

Modélisation des activités

Le modèle des activités permet une identification, exploration et développement des activités [1].

Nous pouvons distinguer deux modèles des activités : le processus de la course ainsi que l'algorithme de sélection du conducteur. Le premier modélise les informations et processus de la course en elle-même, tandis que le second décrit l'algorithme qui permettra le triage des conducteurs en se basant sur plusieurs critères pour pouvoir choisir le conducteur qui correspond le mieux.

Processus de la course

Le modèle des activités de la course est lui-même divisé en deux parties pour une lecture plus simple du modèle.

La première partie peut être décrite comme les activités antérieurs à l'envoi de la course aux conducteurs. Nous pouvons nommés les processus fondamentaux au bon fonctionnement du système : s'inscrire (Processus 1)/se connecter (Processus 2) et s'annoncer en tant que conducteur (Processus 3)/passager (Processus 4). Dès que l'utilisateur a choisi quel rôle iel voulait effectuer dans ce processus (conducteur (Rôle 1) ou passager (Rôle 2)), celui/lle/x-ci se verra demander de remplir des tâches spécifiques à son rôle. Le passager pourra demander une course (Processus 6), renseigner les informations sur la course (Processus 6). Quant au conducteur, iel devra renseigner les informations de son véhicule (Processus 3.5), rester en attente d'une proposition de course (Processus 5), pour finir par recevoir la course et ses informations (Processus 11). Tous deux peuvent en tout temps se déconnecter et ainsi abandonner leur rôle (Processus 7). Le système devra, en parallèle de ces processus, créer l'itinéraire de la course (Processus 9), envoyer la demande aux conducteurs (Processus 10) ainsi que créer une liste des conducteurs (Processus 12).

L'itinéraire de la course sera confectionné à l'aide de l'algorithme d'itinéraire (Ressource NH1), ainsi que les différents différentes API mentionnés précédemment (Ressource Ext1-3).

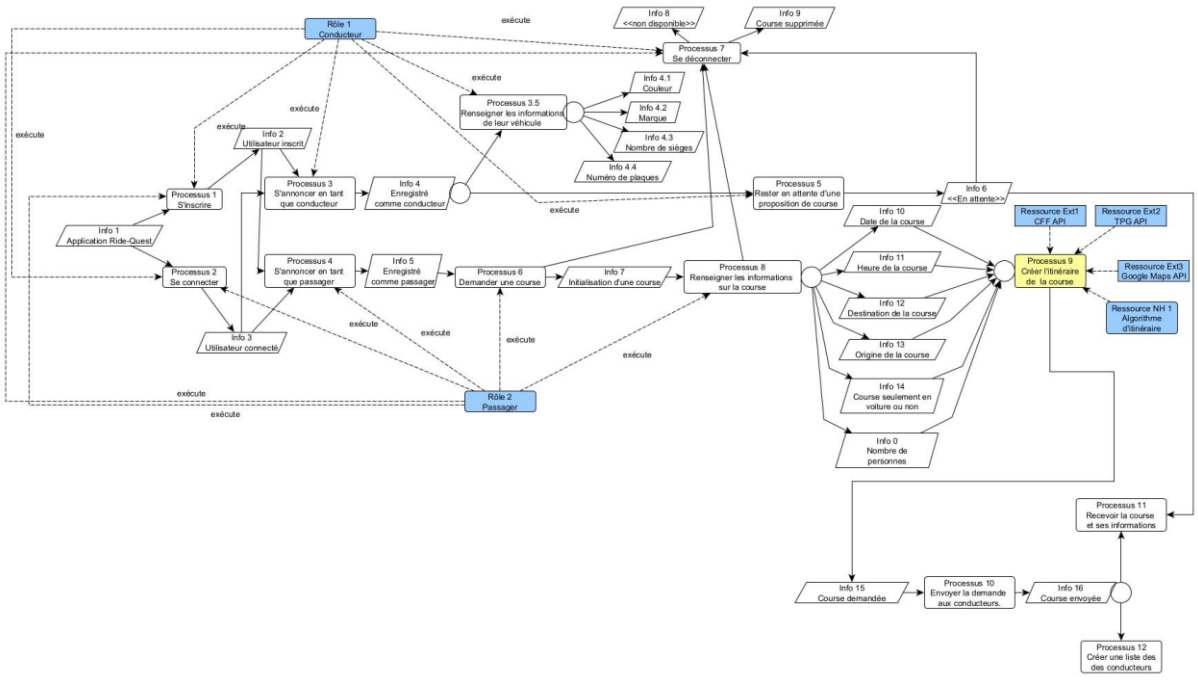

Figure 2: Modèles des Activités de la course - partie 1

Quant à la deuxième partie, celle-ci peut être décrite comme les activités postérieurs à l'envoi de la course aux conducteurs. Les processus sensibles au bon déroulement du système peuvent

encore une fois être mentionnés par rôle. Pour ce qui est du conducteur, ce/tte/x dernier/ère/x peut ignorer (Processus 13) ou accepter la course (Processus 14). Le passager n'a pas de tâches propres à ce rôle à remplir. Iels peuvent tous/e/x deux noter leur binôme (Processus 24) ou le signaler (Processus 25). Pour ce qui est des processus singuliers au système, celui-ci doit inscrire le conducteur dans la liste des conducteurs (Processus 15), choisir un conducteur (Processus 16) (Ressource NH2) (celui-ci sera décrit en détail dans la prochaine section), prévenir le conducteur (Processus 17) et le passager (Processus 18) du matching, supprimer la liste des conducteurs (Processus 19), commencer la course et tenir les statuts à jours (Processus 20-23).

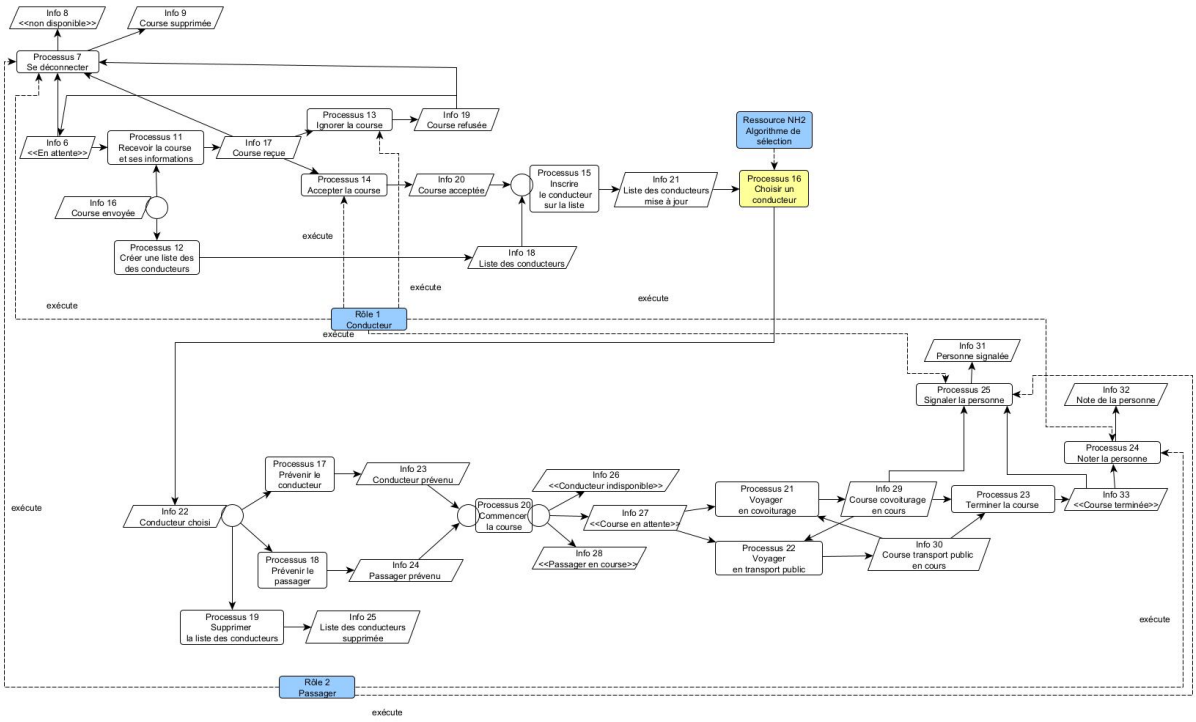

Figure 3: Modèles des Activités de la course - partie 2

Algorithme de sélection du conducteur

Pour revenir sur le processus de sélection d'un conducteur (Processus 16). Celui-ci est fait à l'aide d'un algorithme de sélection du conducteur (Ressource NH2). Nous pouvons le modéliser comme tel. La première étape est d'analyser les informations nécessaires (Processus 26) et ensuite de choisir le conducteur qui convient le plus à ce passager (Processus 27).

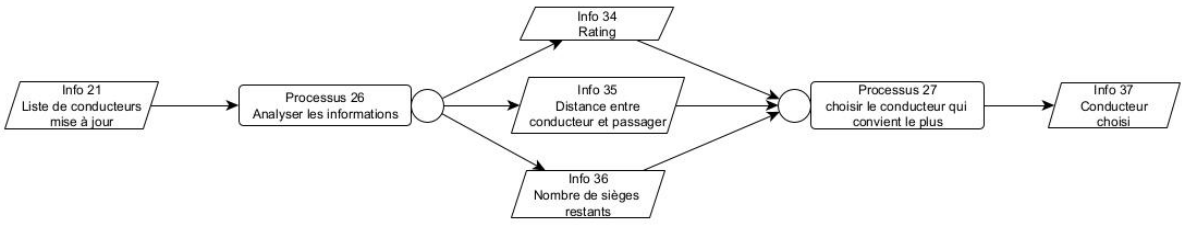

Figure 4: Modèles des Activités de l'algorithme de sélection

Modélisation des règles de gestion

Ce modèle permet une définition des règles de gestions, qui elles-mêmes servent à structurer, coordonner, contraindre, contrôler et influencer les activités de différents services et acteurs [1].

Modèle des objectifs

Les règles de gestion se rattachant au modèle des objectifs viennent au nombre de cinq. Tout d'abord, une règle peut être ajouté à l'Objectif 2 qui stipule que le conducteur ne peut être lié qu'à un seul passager (Règle 1). Une autre règle peut être définie quant à l'Objectif 8 qui prescrit de trouver l'itinéraire le plus efficace (Règle 2). Une troisième règle liée l'Objectif 7 peut être également ajouté qui ordonne que les informations soient valables (Règle 3). L'Objectif 12 se voit suivit d'une règle qui indique que la base de données doit contenir toutes les informations liées aux utilisateurs et aux courses (Règle 4). Une dernière règle peut être mentionnée, qui ajoute à l'Objectif 9 le fait que le matching doit se faire en prenant compte de conditions définies.

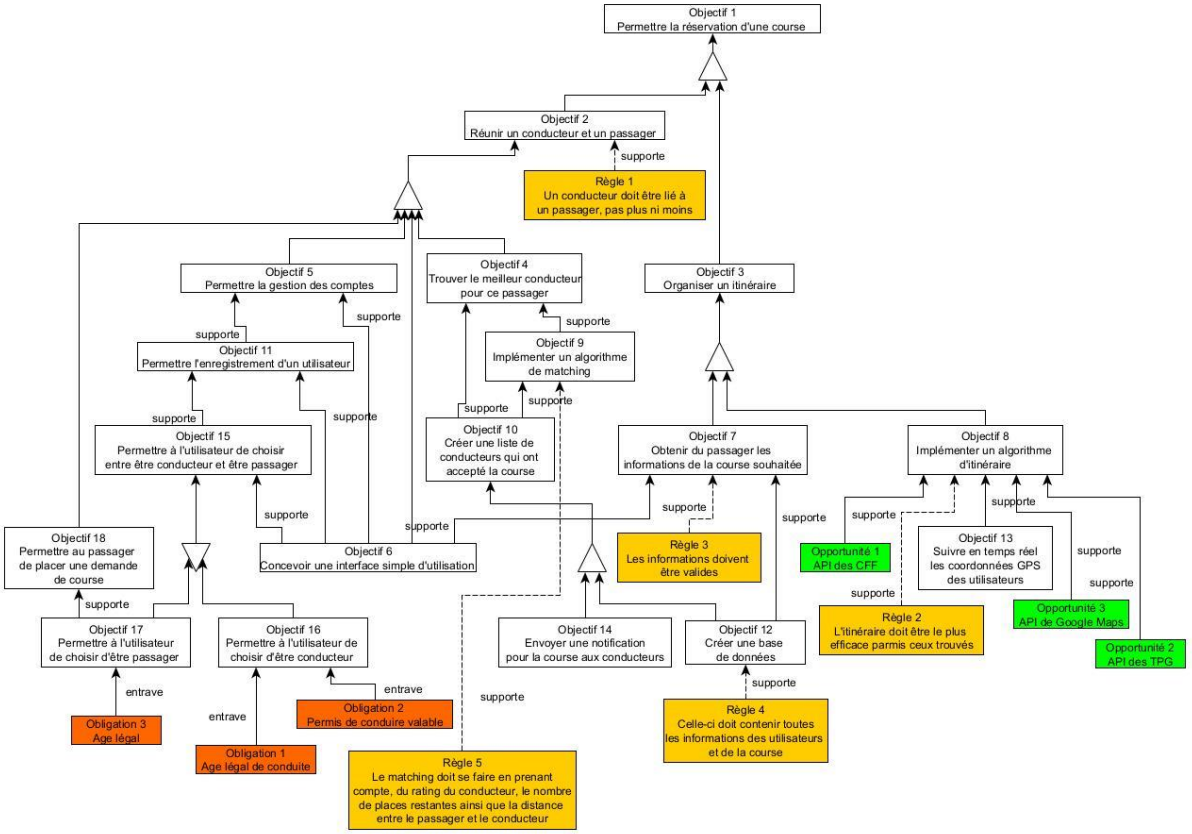

Figure 5: Modèle des Règles de gestion des objectifs

Modèles des activités

Processus de la course

Un total de 19 règles peut être recensé. Les règles rattachées à l'utilisateur en général (Règle 1-3,15-16), au conducteur (Règle 4-5), au passager (Règle 6,12), à la course (Règle 6-8,12-14,17-19) et au système (Règle 9-11).

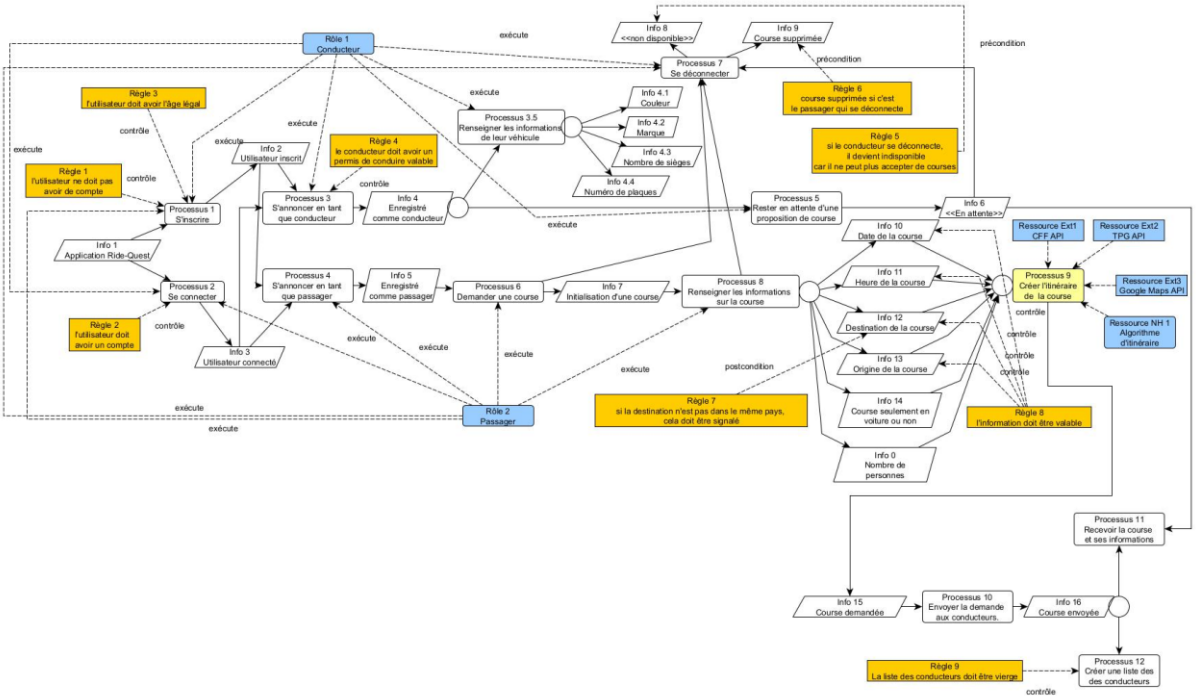

Figure 6: Modèle des Règles de gestion des activités de la course - partie 1

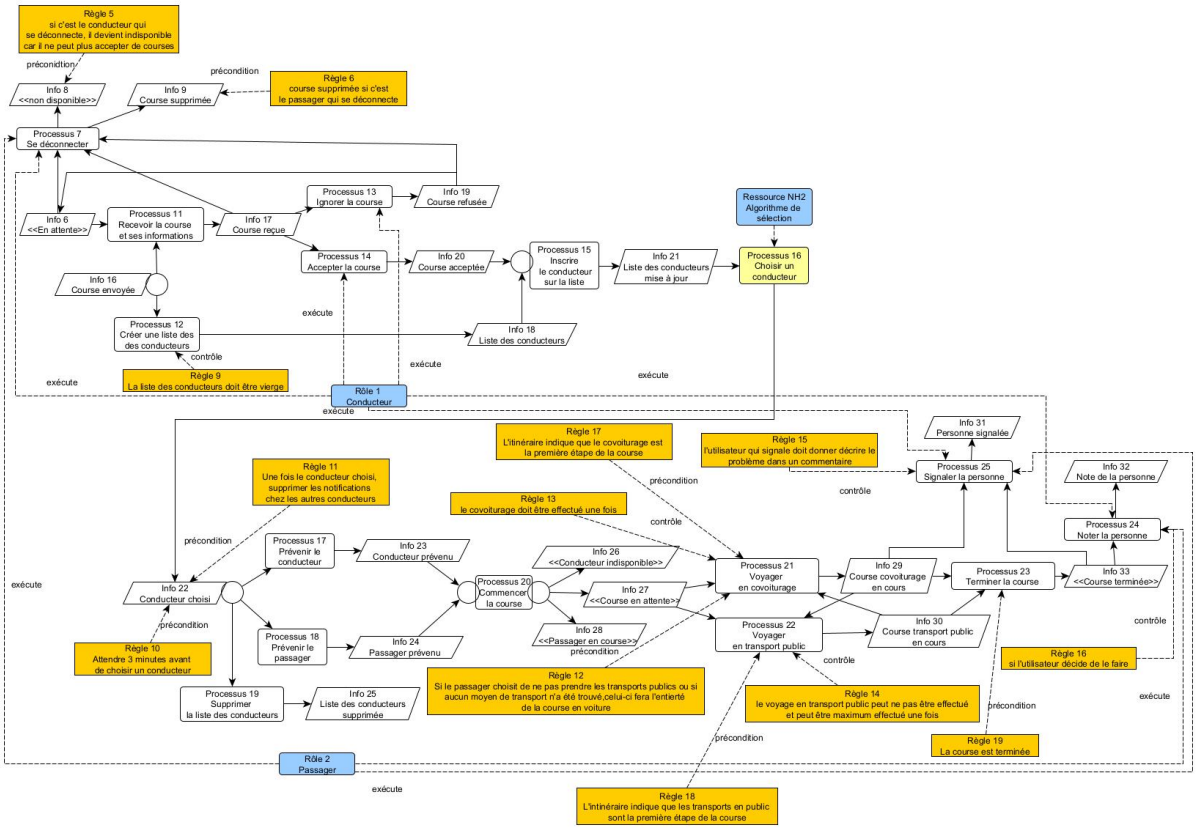

Figure 7: Modèle des Règles de gestion des activités de la course - partie 2

Algorithme de sélection du conducteur

Quant à ce qui est de l'algorithme de sélection du conducteur, deux règles peuvent être mentionnées. La liste des conducteurs peut être mise à jour pendant 3 minutes après l'envoi de la course aux conducteurs (Règle 1), cela pour laisser assez de temps à ces derniers pour se manifester. Une seconde règle définit ce qu'est le rating (Règle 2).

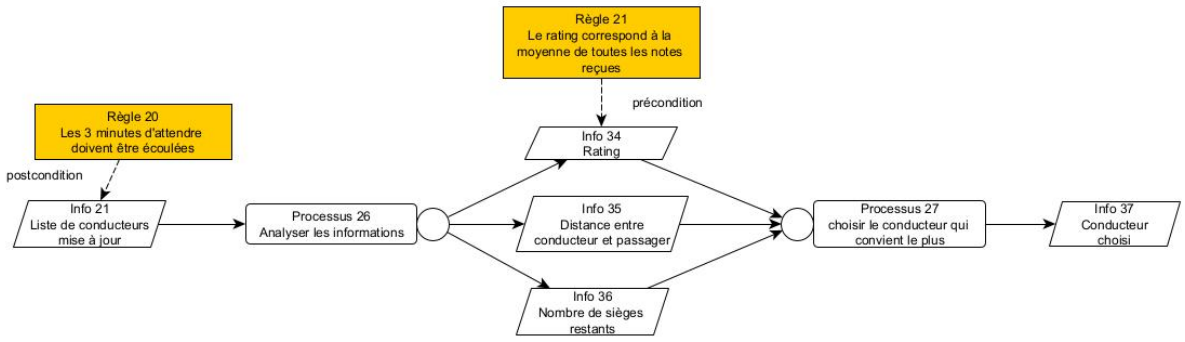

Figure 8: Modèle des Règles de gestion des activités de l'algorithme de sélection

Modélisation des exigences fonctionnelles et non fonctionnelles

Le modèle des exigences permet de décrire les exigences pour le développement d'un service ou d'un système d'information [1].

Beaucoup d'Objectifs SI, d'exigences fonctionnelles et non-fonctionnelles ont été recensés. Nous pouvons en nommer quelques-unes.

L'objectif principal de ce système d'information est le fait de fournir un service de réservation de taxi (Objectif SI1). Celui-ci est accompagné d'une exigence non-fonctionnelles indiquant que le service doit être disponible 24h/24 (Exigence NF1). Cet objectif comprend plusieurs sous-objectifs et exigences : match un passager et un conducteur (Objectif SI2), permettre la gestion des comptes (Objectif SI3) modifier les statuts des utilisateurs et de la course (Exigence F2) ainsi que garder toutes les données dans la base de données (Objectif SI5). Le reste consiste en des objectifs SI, exigences fonctionnelles et non-fonctionnelles servant à appuyer les objectifs SI, les exigences, ainsi que les objectifs et les processus mentionnés dans les modèles précédents.

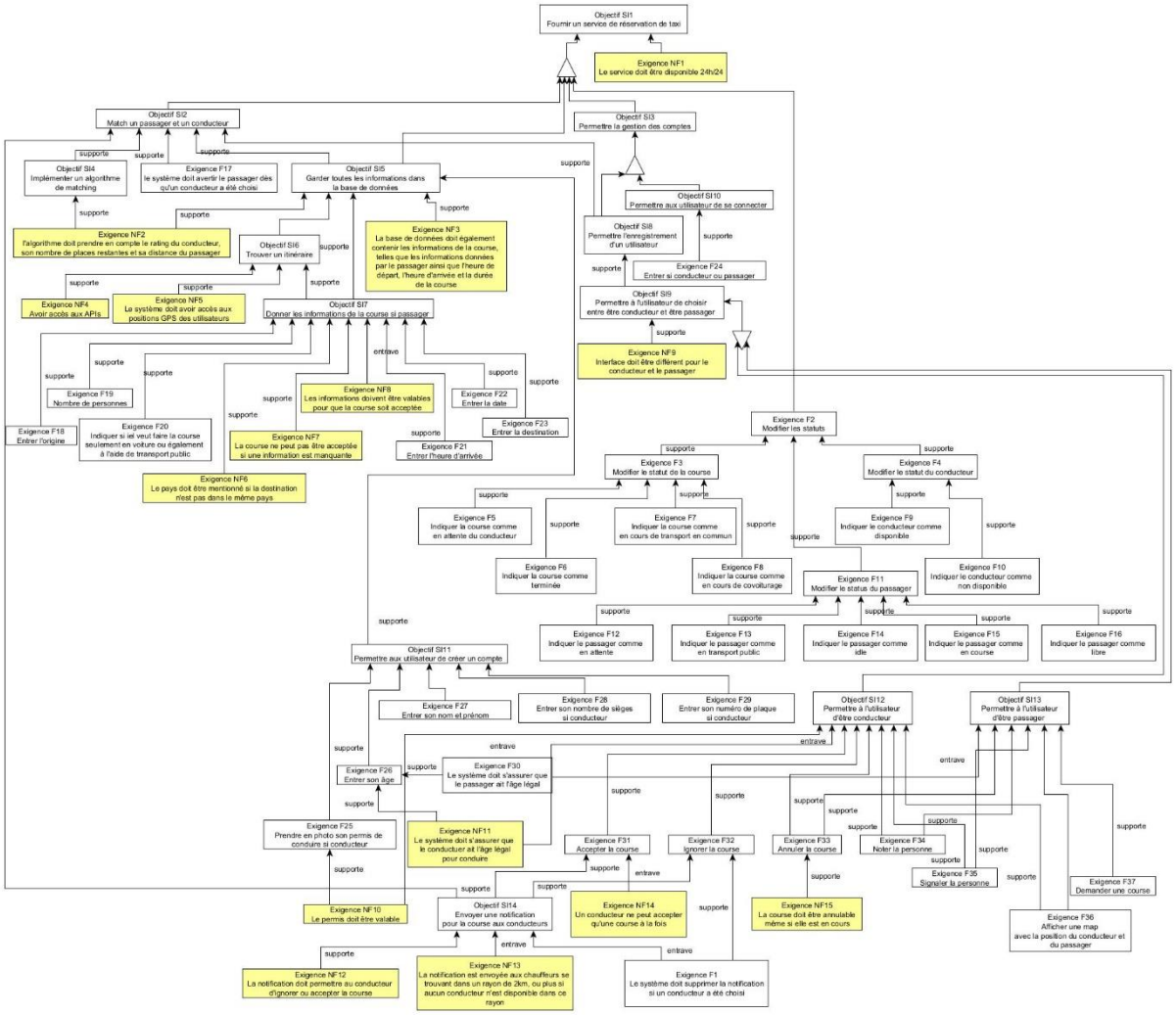

Figure 9: Modèle des Exigences

Use Case (UML)

Le Use Case permet de représenter l'usage que font les utilisateurs du système d'information [1].

Les acteurs sont encore une fois les conducteurs et passagers généralisés en utilisateurs.

Ceux-ci peuvent déclencher plusieurs cas d'utilisation, comme se connecter ou s'enregistrer ainsi que gérer leur compte. Après s'être connecter, l'utilisateur se voit attribuer la possibilité d'offrir ou demander une course, ceci en fonction du rôle qu'iel a choisi. S'iel choisit d'être conducteur, iel offre une course, iel doit dans ce cas donner les informations du véhicule. S'iel choisit d'être passager, iel demande donc une course et doit renseigner les informations de la course souhaitée. Ceci amène un conducteur et un passager à être match et d'ainsi pouvoir compléter la course. L'utilisateur à la possibilité de gérer son compte pouvoir modifier et regarder ses informations.

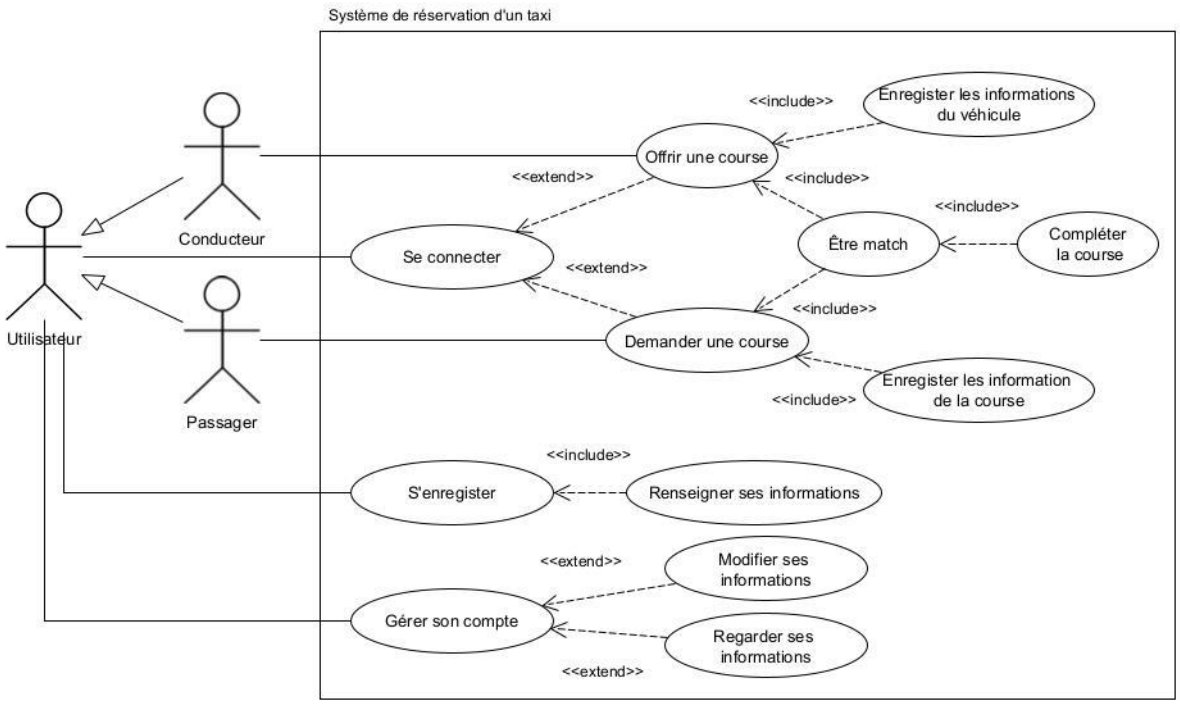

Figure 10: Use Case

Schéma conceptuel

Le schéma conceptuel permet de visualiser les entités et les relations qu'elles ont entre elles. Ce schéma ne sert pas à représenter la base de données, mais à représenter les concepts.

Pour ce schéma conceptuel, 6 concepts peuvent être nommés : utilisateur, conducteur, groupe, voiture, possession et course. A noter que groupe est l'ensemble des passagers pour une course, car le passager faisant la demande pour la course peut en effet voyager avec d'autres passagers. Mais ces passagers n'ont pas de compte et ne sont donc pas enregistré dans la base de données. id_utiliserateur de ce concept est donc l'identifiant du passager qui a fait la demande pour la course. Etant donné que groupe et conducteur sont des spécialisations d'utilisateur, ceux-ci ne portent pas de lien d'association entre eux. Quartes associations font cependant surfaces : conducteur-possession, possession-voiture, conducteur-course et groupe-course. En effet un

conducteur possède une ou plusieurs voitures et une voiture peut être possédée par un ou plusieurs conducteurs (par exemple une voiture familiale), mais comme ceux-ci sont des associations multiples des deux cotées, un concept possession a été rajouté pour savoir qui possède quelle voiture. Ensuite, un conducteur peut offrir entre zéro (si ce dernier est en attente d'une course) et une course et une course ne peut être offerte par qu'un seul conducteur. L'association entre groupe et course fonctionne de la même manière que la précédente, c'est-à-dire qu'un groupe peut n'avoir demandé aucune course pour l'instant mais ne peut en demander qu'une maximum et une course ne peut être demandé que par un groupe.

Nous pouvons également constater les attributs de ces concepts. Les clés primaires sont les attributs clé, qui sont soulignés.

Nous pouvons remarquer deux attributs dérivés, ce qui signifie que la valeur peut être calculée à partir d'autres attributs [1]. Ici, l'âge de l'utilisateur (age_utilisateur') peut être calculé à partir de la date du jour et de la date de naissance de l'utilisateur (date_naiss_utilisateur). La durée de la course (duree_course') peut être calculée à partir de l'heure de début (heure_debut_course) et l'heure d'arrivée (heure_arrivee_course).

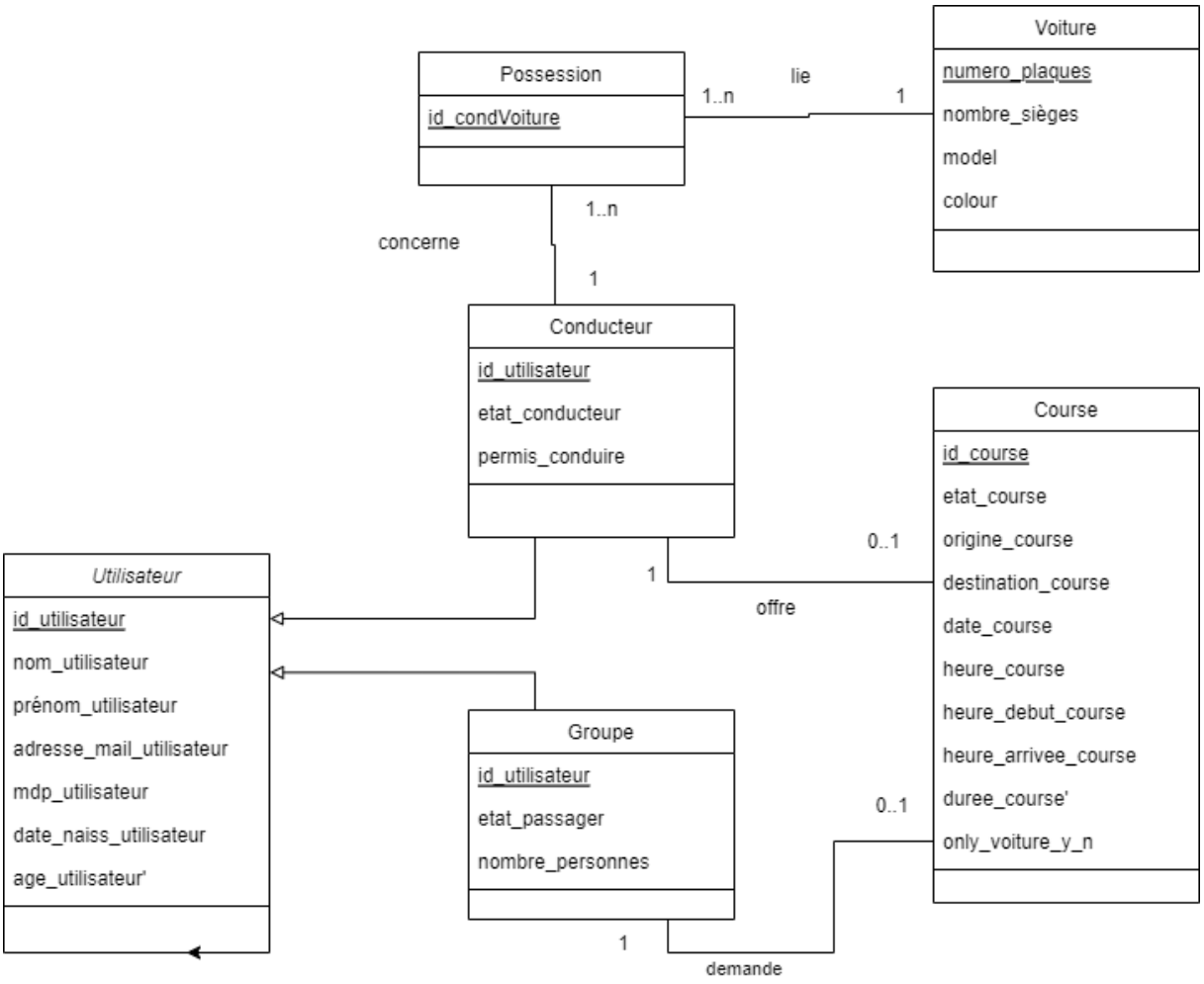

Figure 11: Schéma Conceptuel

Schéma logique

Un schéma logique permet de représenter les clés primaires (Primary Keys) et étrangères (Foreign Keys). Ce schéma représente la base de données et comment ces différentes tables et attributs seront liés et représentés dans celle-ci.

Les clés primaires dépeignent les clés obligatoires de la table et les clés étrangères sont des clés qui se réfèrent à des clés primaires dans une autre table. Une clé peut être une clé primaire et étrangère en même temps.

Dans ce cas, les clés primaires sont les identifiants de chaque table (id_utilisateur, numero_plaques (celui-ci est un identifiant en lui-même car ces numéros sont uniques, clairs, pertinents et monovalués), id_condVoiture, id_course) ainsi que les clés étrangères envoyées faisant références aux clés primaires contenues dans d'autres tables (id_conducteur, id_passager, id_utilisateur et numero_plaques).

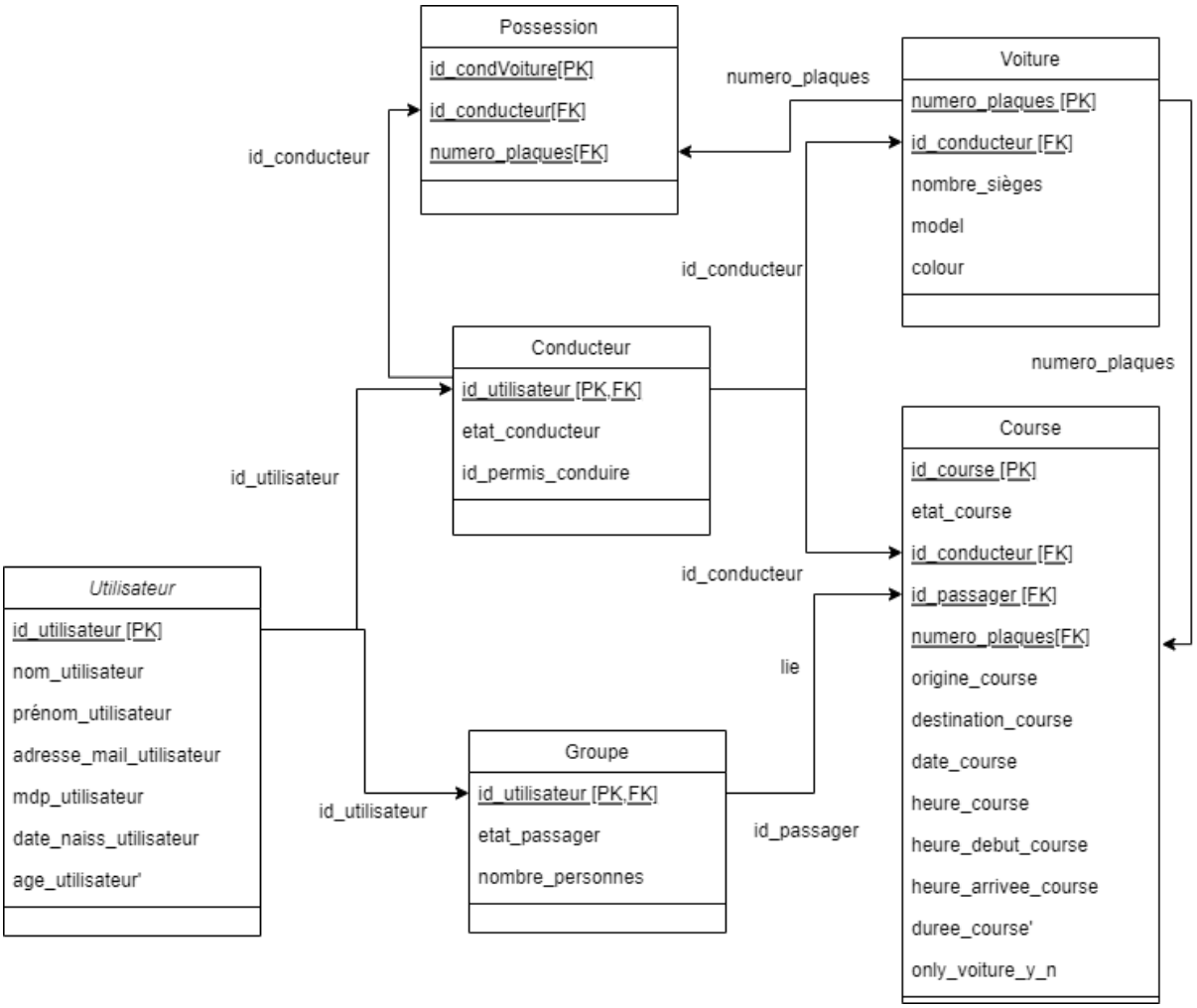

Figure 12: Schéma Logique

Identifier des données de test

Mockaroo sera utilisé pour obtenir des données de test, car cela nous permettra de compiler les données qui nous intéressent. Les données seront celles décrites dans le schéma logique.

Usage des outils

Nous n'avons pour l'instant pas utilisé ni Gitlab, ni phpMyAdmin, n'ayant pas encore de parties implémentées ni de base de données concrète.

Draw.io et yEd ont été utilisé pour faire les graphes d'analyse des objectifs ainsi que les schéma conceptuel et logique. yEd a permis de faire les modèles, tandis que Draw.io a été utilisé pour faire les schémas conceptuel et logique, les outils proposés par yEd étant moins adéquat pour remplir cette tâche que ceux mis à disposition par Draw.io.

Mockaroo va, comme mentionné précédemment, être utilisé pour compiler des données de test.

Conclusion

Dans ce rapport, nous avons introduit le projet en mentionnant ses problématiques, présenté les différents modèles d'analyse des objectifs, les schémas conceptuel et logique, identifier des données de test, ainsi qu'énuméré les outils utilisés tout au long de ce dernier.

Les modèles nous donnent une base robuste pour mener ce projet à bien, l'implémentation nécessitant une connaissance des objectifs et de ce qui doit être réalisé pour ne pas implémenter quelque chose d'erroné.

Nous suivrons ce rapport par un deuxième rapport, qui prendra en compte les retours de nos clients quant aux améliorations à effectuer et la suite logique de celui-ci.

Références

[1] Prof. Jolita Ralyté, cours « Analyse des objectifs »