Année 2018-2019

RAPPORT DE PROJET *–*

BASE DE DONNÉES

*Projet réalisé par*

Nabil Aissat

Cessna Jefferson

Oleksandr Firsov

Emre Aydin

Sommaire

Introduction …………………………………………… p. 3

I- Reflexion ………………………………………….... P. 3

II- Organisation …………………………………….... P. 3

1. Diagramme de Gantt ……………………………. P. 3
   1. Previsionnel …………………………………. P. 4
   2. Réel ………………………………………. P. 4
2. Modélisation du schéma de base de données ... P. 4

III- Phases de développement ………………………. P. 6

1. Interface Client …………………………………. P. 6
2. Interface Superviseur ……………………………. P. 8

V- Modélisation …………………………………………. P. 8

VI- Difficultés ……………………………………………. P. 12

VII- Conclusion ………………………………………. P. 13

Annexes ……………………………………………. P. 15

Introduction

Dans le cadre de nos études à l’université de Grenoble-Alpes, en licence MIAGE, nous avions comme projet de développer une application en java afin de répondre au mieux aux besoin de la ville de Mahingan Falls. Cette dernière souhaite se doter d’un service de vélos en libre-service sur le modèle du Vé’lib parisien. Ce service qui se nommera donc “VéPick”. Nous sommes chargés du système de gestion de données de VéPick, basé sur une base de données relationnelle.

Cela implique donc de gérer à la fois le parc de vélos, les stations et les clients. Deux

applications doivent permettre d'accéder à la base de données : une pour les clients et une pour les superviseurs du réseau. En effet, nous avons développé deux applications : une pour le superviseur pour la gestion de ce service et une pour le client afin qu’il puisse bénéficier pleinement de ses services.

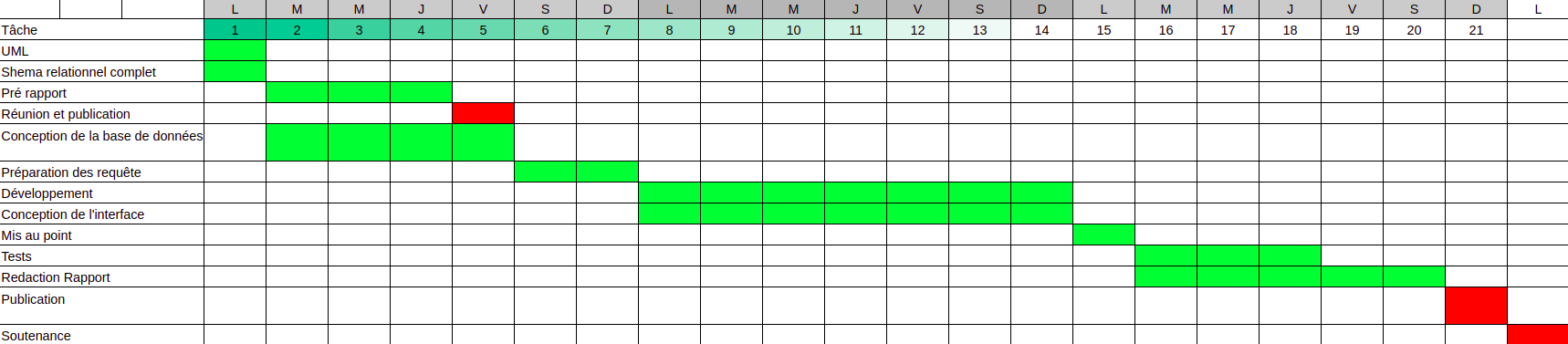
I - Organisation

1. Diagramme de Gantt

a) Prévisionnel

Afin de répondre au mieux aux besoins du client dans les temps impartis, il a été nécessaire d’être organisé au mieux.

Pour cela nous avons réalisé un diagramme de gantt en listant l’ensemble des tâches à réaliser pour arriver à la finalité du projet.



Nous nous sommes donc basés sur 21 jours. La première semaine concernait la réalisation de l’UML ainsi que du Schéma relationnel afin de présenter un premier rendus aux enseignants compétents.

Formant un groupe de 4 étudiants, nous avons divisés les besoins demandées par 4, ou 2 besoins par personnes. Chaque personne devait préparer ses requêtes pour la fin de la semaine 1.

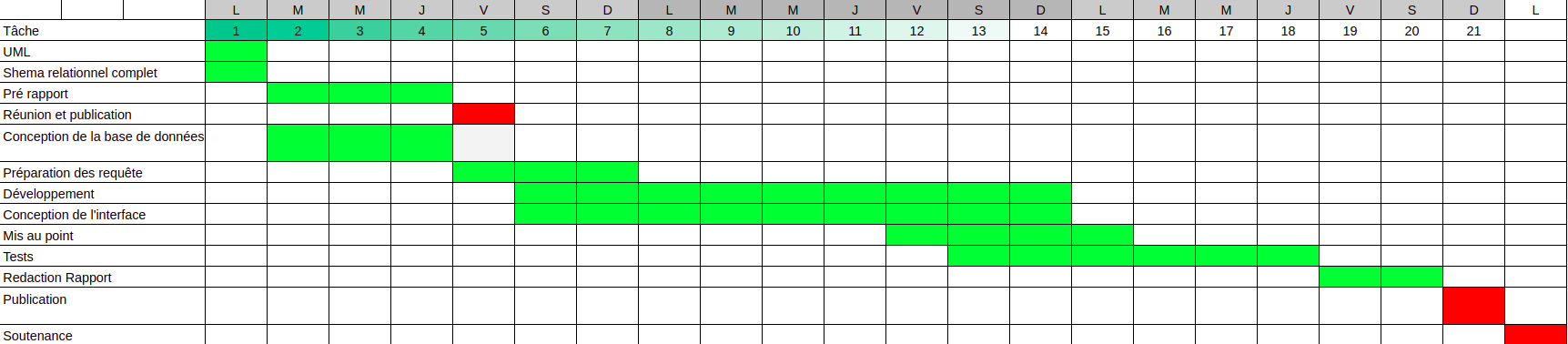
Mais parallèlement nous nous sommes réunis à plusieurs reprises afin de réaliser la base de données. L’objectif était donc de préparer les tâches de la semaine suivante. Après avoir installé la base de données et avoir ajouté des données dans cette dernière.

Requêtes préparées, nous avons donc commencé à développer les fonctions en java. Pour cela nous avons donc partagé un projet sur notre git pour que chacun de nous ait accès au projet de façon instantanée.

Il ne faut pas oublier l’interface qui est développée parallèlement par une personne de notre équipe. Après une semaine de développement, nous nous sommes réunis pour faire la mis au point de l’ensemble de nos fonctionnalités.

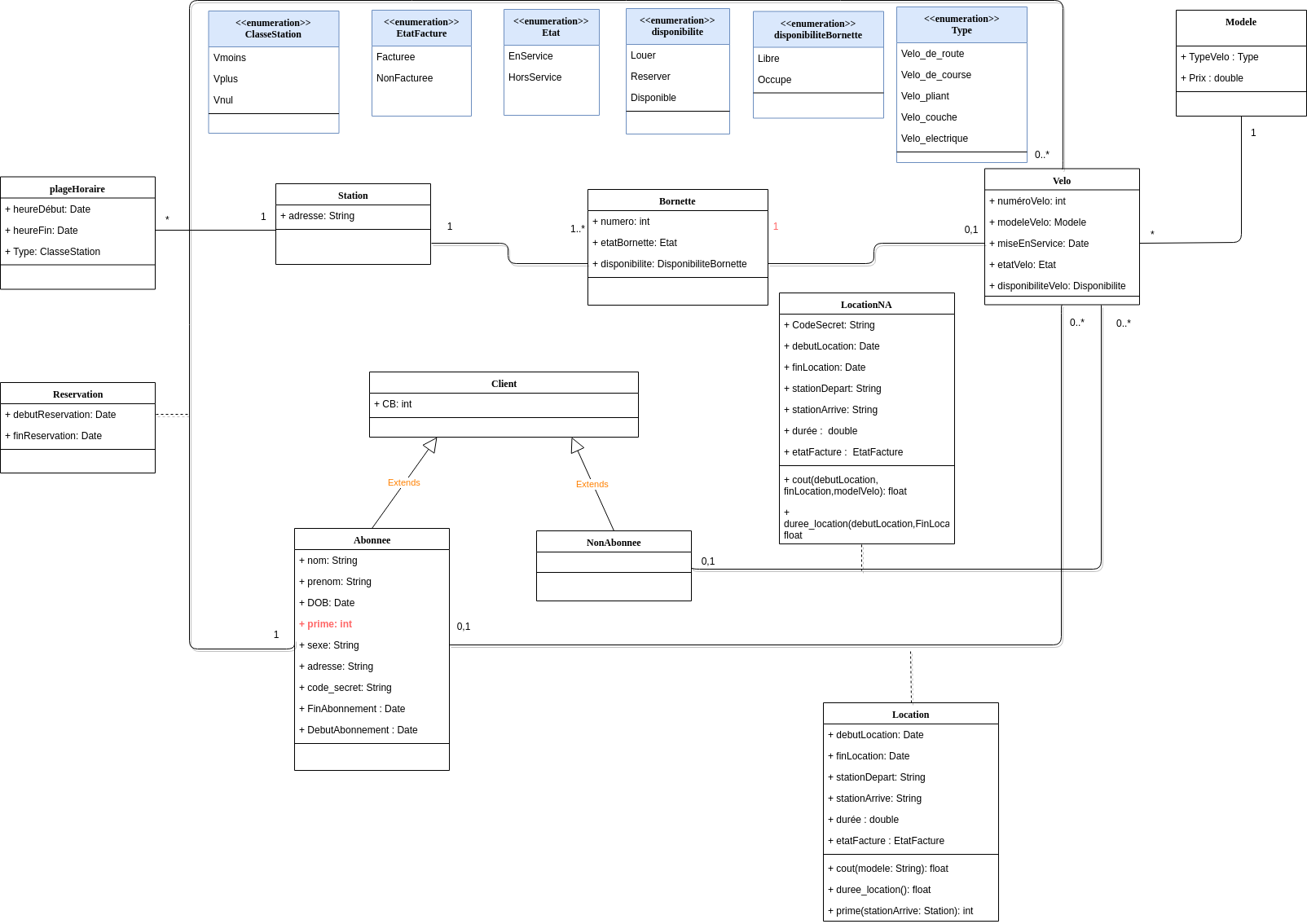
Et enfin, nous avons établis un ensemble de tests pour gérer les exceptions etc..

b) Réel



Nous avons réussi à développer toutes les fonctionnalités que le client attend. Cependant certaines tâches ont durée plus longtemps que prévu et d’autres moins.

II - Modélisation du schéma de la base de données



Location abonné et Location non abonné nous avons ajouté un etat “EtatFacture” qui illustre si une facture est facturée ou non facturée.

III - Phase de développement

1. Interface client

Nous commençons à développer l’ensemble des fonctionnalités qui sont susceptibles d’être exécutées par le client.

La toute première fonctionnalité que l’on a développé est destinée à l’authentification de la personne abonnée ou non.

Une personne qui est déjà abonnée au préalable s'authentifie avec son nom, prénom et son code secret. Quant à la personne non-abonnée, on parlera d’une authentification lorsque ce dernier souhaite rendre son vélo. Cette option est plus détaillée dans la suite de ce présent rapport.

L’abonnement au service : Une personne pourra s’abonner au service en inscrivant son numéro de code bancaire après s’être authentifiée.

|  |  |
| --- | --- |
| Emprunt d’un vélo | |
| Personne abonnée | Personne non-abonnée |
| Cette personne s'identifie dans le système pour procéder à un emprunt. Aucun paiement se fait pendant ce temps mais le système enregistre l’historique de toutes les démarches faites par chaque utilisateur.  Le paiement se fait lorsque cette dernière rendra le vélo | Une personne non-abonnée pourra aussi emprunter un vélo sauf que celui-ci n’est pas dans l’obligation de s'identifier. Elle aura seulement à emprunter le vélo et préserver bien précieusement le code généré par le système qui est unique. Ce code servira pour rendre le vélo et donc pour le paiement. |

|  |  |
| --- | --- |
| Rendu d’un vélo | |
| Personne abonnée | Personne non-abonnée |
| Après authentification, la personne peut rendre son vélo. Cette dernière doit seulement cliquer sur “rendre” affichage des stations libres puis bornettes libres puis il tape le numero de vélo  Le système regarde si le vélo existe dans la base de données et valide ou non en fonction des résultats. | Après avoir cliquer sur “rendu” la personne non-abonnée pourra visualiser l’ensemble des stations libres puis en choisir une. Après ceci, il fera de même pour une bornette libre et indiquera le code sur le vélo ainsi que son code secret pour la validation du rendu. |

Toutes les fonctionnalités liées à l’abonnement au services, aux emprunts ainsi qu’aux rendus de vélos

A tout moment, le client peut visualiser l’ensemble des bornes Vplus et Vmoins. Cela permet aux clients abonnées de visualiser les bornes à laquelle il pourraient déposer leur vélo afin de bénéficier d’une quelconque prime.

Le client a la possibilité d’alerter le système si le vélo est endommagé. Attention ici, nous gérons l’exception temporel. Si l’utilisateur alerte le système trois minutes après la réservation, les dégâts seront facturés de façon automatique.

|  |
| --- |
| Calcul la durée de location |
| A savoir que chaque modèle de vélo possède un tarif différent. Nous stockons dans la base de données le prix par demi-heure par modèle. En fonction de ceci, nous arrivons à calculer le prix de la location en prenant en compte la date de début ainsi que de fin de location. |

Les procédures développés :

* AjoutLocationAbonne

Cette procédure nous permet de créer une nouvelle location d’un abonné

**Annexe - Procédure AjoutLocationAbonne**

* updateEtatVeloBornetteLocation

Cette procédure nous permet de mettre à jours l’état d’un vélo ainsi que de la bornette au moment de la location

**Annexe - Procédure updateEtatVeloBornetteLocation**

* AjoutClientNonAbonne

Cette procédure nous permet d’ajouter dans la base de données un client non abonné

**Annexe - Procédure AjoutClientNonAbonne**

* AjoutLocationNonAbonne

Cette procédure nous permet de créer une nouvelle location d’un non-abonné

**Annexe - Procédure AjoutLocationNonAbonne**

* updateEtatVeloBornetteRendu

Cette procédure nous permet de mettre à jours l’état d’un vélo ainsi que de la bornette une fois que cette première soit rendue.

**Annexe - Procédure updateEtatVeloBornetteRendu**

Trigger crée :

* contrainteReservations : ce trigger gère le non-recouvrement. C’est à dire qu’une personne ne peut réserver un vélo de 8h à 10h puis un autre de 9h à 11h. Un client ne peut donc pas avoir plusieurs réservations avec un recouvrement

**Annexe - Trigger**

1. Interface superviseur

Sur cette interface, le superviseur a la possibilité de visualiser le nombre de vélos dans chaque stations (non endommagés), le nombre de vélos endommagés puis le nombre de places libres.

Et puis, il peut visualiser et modifier les plages horaires Vmois & Vplus. Pour cela, il sélectionne le numéro de la plage horaire à modifier (après affichage) puis le système lui proposera de modifier ou non l’heure de début, de fin, le type ainsi que le numéro de station à laquelle celui-ci est rattachée.

IV- Modélisation

SCHÉMAS RELATIONNEL :

Station(**numStation**,adresse)

(NS, AD) ∈ Station ⇐⇒ la station identifiée par **NS** dont l’adresse est **AD**.

Bornette(**numBornette**, etatBornette, disponibilite ,numStation, numVelo)

(NB, EB, NS,NV) ∈ Bornette ⇐⇒ la bornette identifiée par **NB** est dans un état **EB** se situe à la station identifiée par **NS** et accueille le vélo numéro **NV** .

Velo(**numVelo**,numModele,MiseEnService,EtatVelo,DisponibiliteVelo)

(NV,MV,DS,EV,DV) ∈ Velo ⇐⇒ le vélo de numéro **NV** a comme modèle **MV** est mise en service à la date **DS** et dans un état **EV** et comme disponibilité **DV**.

ClientAbonne(**numClientAbonne**,Nom,Prenom,DateDeNaissance,Sexe,Adresse,NumCB,CodeSecret,Prime,DebutAbonnement,FinAbonnement)

(NC,N,P, D,S,A,NumCB,code,pr,debut,fin) ∈ ClientAbonne ⇐⇒ le client de numéro **NC** , nom **N** , prenom **P**,date de naissance **D**, sexe **S**, adresse **A** ,possède la carte bancaire numéro **NumCB** , un code secret **code** ,une prime **pr,** un abonnement de la date **debut** jusqu'à la date **fin** .

ClientNonAbonne(**numClientNonAbonne**,NumCB)

(NClientNA, NumCBl) ∈ ClientNonAbonne ⇐⇒ le client de numéro **NClientNA** possède la carte bancaire numéro **NumCBl**.

LocationAbonne(idLocationAbonne,DebutLocation,FinLocation,Duree,StationDepart,StationArrivee,EtatFacture)

(ID, NC,NV,DL,DF,D,SD,SA,EF) ∈ LocationAbonne ⇐⇒ La location abonnée est identifiée par **ID** effectué par le client abonné numéro **NC** qui a louer le vélo numéro **NV** a partir de la date **DL** jusqu'à la date **DF** pour une durée **D** de la station **SD** et il a rendu à la station **SA** et un état de facturation **EF**.

LocationNonAbonne(idLocationNonAbonnee,DebutLocation,FinLocation,StationDepart,StationArrivée,CodeSecret,EtatFacture)

(ID, NC,NV,DL,FL,D,SD,SA,CS,EF) ∈ LocationNonAbonne ⇐⇒ La location non abonnée est identifiée par **ID** effectué par le client non abonné numéro **NC** qui a louer le vélo numéro **NV** a partir de la date **DL** jusqu'à la date **FL** pour une durée **D** de la station **SD** et il a rendu à la station **SA** et possède le code secret CS et un état de facturation **EF**.

Reservation(idReservation, numClientAbonne,numVelo,DebutReservation,FinReservation)

(ID, NC,NV,DR,FR) ∈ Reservation ⇐⇒ La réservation est identifiée par ID effectué par le client abonné numéro **NC** a réserver le vélo numéroNV a partir de la date **DR** jusqu'à la date **FR .**

PlageHoraire(**numPlageHoraire**,HeureDebut,HeureFin,Type, numStation)

(NP,HD, HF,T,NS) ∈ PlageHoraire ⇐⇒ la plage horaire identifiée par **NP** est de type **T** entre heure début **HD**, une heure fin **HF** et un numéro de station **NS**.

Modele(**numModele**,TypeVelo,Prix)

(NM, T, P) ∈ Modele ⇐⇒ le modele identifiée par **NM** est du type **T** et a comme prix de location **P .**

Description des domaines:

dom (numStation) = [1 ,.. ,999]

dom (numBornette) = [1 ,.. ,99[

dom (EtatVelo) = dom (etatBornette) = { “EnService”, “HorsService”}

dom (numStation) = [1 ,.. ,99]

dom (numVelo) = [1 ,.. ,999]

dom (ModeleVelo) = {“Gtx325”, “VTT100” ,…}

dom (MiseEnService) = date { données à la granularité du jour. }

dom (DisponibiliteVelo) = {“Louer” ,”reserver” ,”disponible”}

dom (clientAbonne) = dom (clientNonAbonne) = [1 ,.. ,999999]

dom (Nom) = {Aissat, Aydin, Firsov, Jefferson...}

dom (Prenom) = {Nabil, Emre, Oleksandr, Cessna ,...}

dom (DateDeNaissance) = [1900 , ∞[

dom (Sexe) = {“H”, “F”}

dom (Adresse) = {“12 Rue Charlemage Grenoble 38000” ,...}

dom (NumCB) = [0 ,.. ,9999999999999999]

dom (CodeSecret) = {“FpOt89V” ,”89DaeDX” ,...}

dom (Prime) = [1 ,.. , 59[

dom (DebutAbonnement) = dom(FinAbonnement) = date { données à la granularité du jour. }

dom (Duree) = [1 ,.. , 999]

dom (StationDepart) = dom (StationArrivee) = {“Grenoble” ,”Saint-Martin” ,...}

dom (DebutReservation) = dom (FinReservation) = date { données à la granularité du jour. }

dom (EtatFacture) = {Facturee, NonFacturee}

dom (numPlageHoraire) = [1 ,.. ,99]

dom (HeureDebut) = dom (HeureFin) = date { données à la granularité du jour. }

dom (numModele) = [1 ,.. , 999]

dom (Type) = {Vplus, Vmoin, Vnull}

dom (Prix) = [0.0 ,.. ,99.0]

Les contraintes d’intégrité référentielle:

**Dans la table PlageHoraire:**

PlageHoraire[numStation] **⊂** Station[numStation]

**Dans la table Bornette:**

Bornette[numStation] **⊂** Station[numStation]

Bornette[numVelo] **⊂** Velo[numVelo]

**Dans la table LocationAbonnee:**

LocationAbonnee[numClientAbonne] **⊂** ClientAbonne[numClientAbonne]

LocationAbonnee[numVelo] **⊂** Velo[numVelo]

**Dans la table LocationNonAbonnee:**

LocationNonAbonnee[numClientAbonne] **⊂** ClientNonAbonne[numClientAbonne]

LocationNonAbonnee[numVelo] **⊂** Velo[numVelo]

**Dans la table Reservation:**

Reservation[numClientAbonne] **⊂** ClientAbonne[numClientAbonne]

Reservation[numVelo] **⊂** Velo[numVelo]

**Dans la table Velo :**

**Velo[numModele] ⊂ Modele[numModele]**

IV - Difficultés

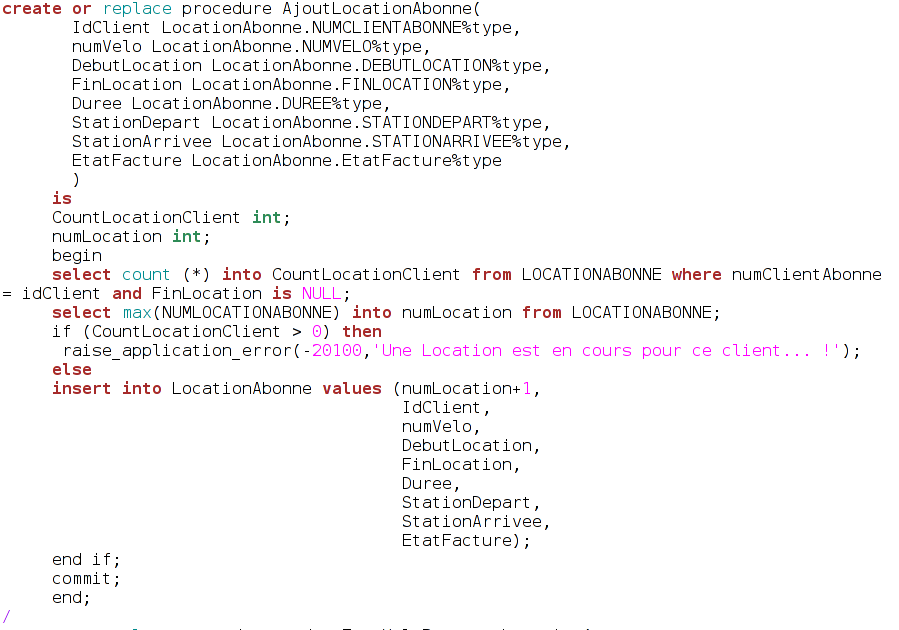
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nature de risque | Description | Gravité 1-5 | Impact | Actions préventives | Actions corretrices |
| Technique | Problème de réseau machine en salle TP | 1 | Retard sur le travail et perturbation du planning | Chacun apporte son ordinateur et avoir au moins une connexion wifi (via Tél.) | Signaler le service informatique et utiliser nos ressources personnelles |
| Communication | Développement de deux fonctions identiques ! | 2 | Redondance dans notre travail, retard engendré | Création d’un drive pour le travail partagé et un groupe de discussion sur FaceBook | Organiser une réunion pour rectifier ce problème de communication |
| Organisation | Non respect des délais | 3 | Retard sur le travail, risque de replanification | Nouvelles attributions de tâches | Réaffectation de la tâche concernée et attribution d’une autre tâche à la personne |
| Technique | Mauvaise conception de la modélisation | 2 | Travail erroné | Valider (par équipe) la modélisation avant d’avancer dans le pojet | Reprendre la modélisation. Priorisation des tâches, augmentation du coût en heure de travail |

VI - Conclusion

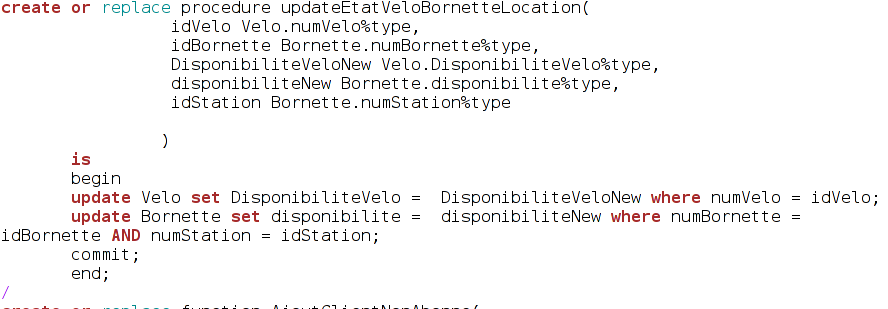
Pour conclure, nous avons effectué ce projet avec beaucoup de sérieux et de motivation. Cela s’est révélé comme un veritable defi pour l’ensemble de notre équipe. En effet, les débuts n’ont pas été facile lorsque nous devions coordonner les tâches et planifier l’organisation du projet. Mais nous avons réussi à relever ce défi en déployant de nouveaux outils de communication afin d’améliorer celle-ci.

Annexes :

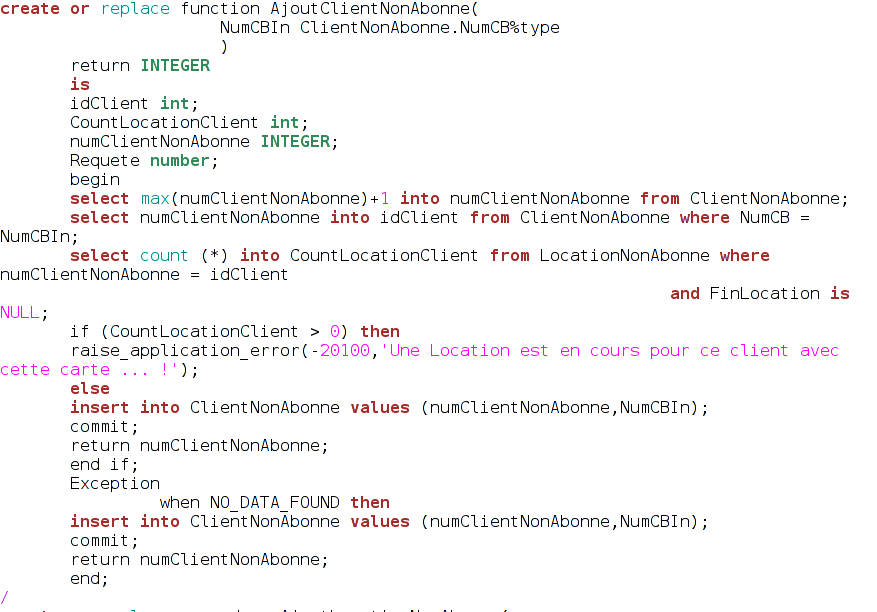
**Annexe - Procédure AjoutLocationAbonne**

****

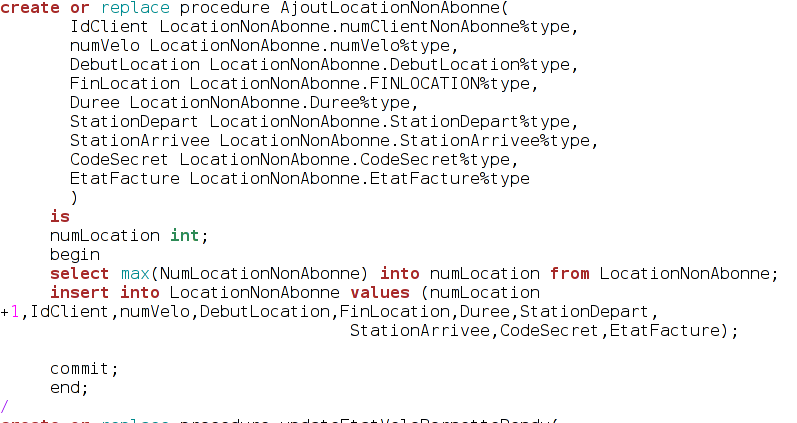
**Annexe - Procédure updateEtatVeloBornetteLocation**



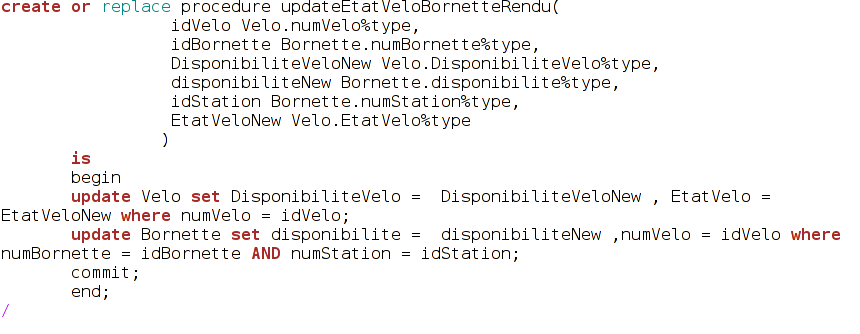
**Annexe - Procédure AjoutClientNonAbonne**



**Annexe - Procédure AjoutLocationNonAbonne**



**Annexe - Procédure updateEtatVeloBornetteRendu**



**Annexe - Trigger**

