

Agence ESCAPADE

Le document ci-dessous décrit en version simplifiée le fonctionnement de l'agence de voyage Escapade propose sur Internet un service de conciergerie pour l'organisation de séjours de courte durée (week-end) à Paris intra-muros.

Un séjour est défini par un thème (sur 3 caractères) et une date. A chaque thème correspond un arrondissement de Paris et à chaque date (uniquement des week-ends), un numéro de semaine.

Un séjour comprend la mise à disposition d'un véhicule pour les deux journées (samedi et dimanche) et l'hébergement dans un logement RBNP organisation avec laquelle Escapade a un contrat global.

Elle utilise, pour cette activité, ses 40 voitures (cabriolets 2 places ou berlines 4 places) qui sont louées le week-end uniquement (samedi et dimanche). Ces véhicules sont des véhicules de luxe dont l'état est suivi de très près. Une voiture, c'est marque, modèle, immatriculation et toute autre information qui vous apparaîtra utile ou pertinente.

L'hébergement est choisit par l'agence parmi les offres RBNP de l'arrondissement correspondant au thème choisi.

Pour un séjour, l'agence se charge de fournir au client une voiture (pour les deux journées de samedi et dimanche) disponible le samedi matin dans le parking de l'arrondissement concerné par le thème choisi et un logement pour la nuit de samedi à dimanche. La voiture sera rendue par le client le dimanche soir au plus tard, dans le parking de son choix parmi les parkings en relation avec Escapade.

A la fin du week-end, le séjour est noté par le client : note de 0 à 5 et appréciation textuelle.

Gestion des véhicules

Chaque véhicule est affecté à un contrôleur (ils sont 3) qui veille au parfait état de chacun des véhicules sous sa responsabilité.

- Les véhicules non loués sont stationnés sur des places réservées dans des parkings parisiens. L'agence Escapade a un contrat global de location de places de parking qui lui attribue des places réservées (et identifiées de A0, réservée pour les départs, à A9) dans 20 parkings parisiens (1 par arrondissement) plus 1 à Roissy et 1 à Orly. La liste des parkings vous est fournie ci-joint.
- Lorsqu'une voiture est louée, le client la récupère sur la place qui lui a été indiquée sur le mail de confirmation du week-end et la rapporte sur n'importe laquelle des places réservées à Escapade. Il envoie par message à Escapade la position de la voiture louée (parking+ numéro de place), ce qui marque la fin de sa location
- Quand une voiture rentre de location, elle n'est plus disponible à la location tant que le contrôleur affecté ne l'a pas vérifié et n'a pas revalidé sa disponibilité.
Cela se fait normalement rapidement en quelques jours et dès le mercredi suivant la location, la voiture est en général redevenue disponible.
Si une voiture est indisponible, le motif de son indisponibilité doit être renseigné.

Toute intervention effectuée par le contrôleur, aussi minime soit-elle (mécanique, mais aussi nettoyage intérieur ou extérieur, plein d'essence), doit être enregistrée, mémorisée et décrite dans le SI de la société. Il peut être demandé au contrôleur de déplacer une voiture d'un parking à un autre.

Gestion des logements

L'agence utilise les services de RBNP qui propose des locations de courte durée dans Paris. A la réception d'une demande de séjour par un client, Le SI de l'agence envoie une demande à RBNP.

RBNP est une organisation mondiale qui gère des dizaines de milliers de logements. Son système informatique est basé sur une base de données noSQL à laquelle on ne peut accéder directement.

RBNP met à la disposition de ses clients une API d'accès.

- RBNP loue ses logements par semaine et ils sont donc disponibles ou non semaine par semaine.
- On envoie à RBNP un message (au format JSON pour les bd noSQL) précisant la ville à laquelle on s'intéresse et le numéro de semaine. **(*J1)**
- On reçoit en réponse un message JSON décrivant la liste des logements disponibles pour ce numéro de semaine (vous trouverez en annexe un exemple d'un tel message). **(*J2)**

Gestion de la relation client

Un client choisit un séjour défini par un thème et une date. Pour cela, il remplit le formulaire du site et pour finaliser sa demande il précise son numéro d'inscription s'il est déjà client et sinon son nom, son adresse (en version atomique – 1 seul champ de caractères -) son numéro de tel, son email.

Si le demandeur est un nouveau client le système informatique l'enregistrera comme nouveau client et lui attribuera un numéro client (un numéro séquentiel d'inscription est suffisant).

La construction du site web ne fait pas partie de ce qui vous est demandé

Pour chaque demande client

La validation du formulaire provoque l'envoi d'un message structuré, format XML **(*M1)**, vers le serveur de l'agence. A la réception de ce message et après sa lecture **(*P1)**, le programme du serveur évalue la faisabilité de ce séjour.

- Une voiture est-elle disponible pour la semaine choisie et laquelle ? **(*R1)**
- Un logement est-il disponible dans l'arrondissement correspondant à la réservation ? **(*J3)**

Si la réponse est oui à ces deux questions le séjour est réalisable.

On sélectionne de préférence une voiture se trouvant dans un parking de l'arrondissement correspondant au thème choisi.

S'il n'y en a pas toute autre voiture dans un autre parking fera l'affaire. On enverra alors au client **le nom du parking correspondant** au thème choisi et **comme numéro de place, la place A0**.

Dans ce cas, après confirmation du séjour, un message SMS sera envoyé au contrôleur de la voiture pour lui demander de transférer la voiture dans le parking indiqué (l'envoi des messages SMS sort du champ de vos responsabilités) et il aura en charge de mettre à jour la position de la voiture.

Si le séjour est réalisable, un numéro de séjour est attribué au séjour défini par le client, le séjour est enregistré dans la BD (en position séjour non confirmé) et un mail de **confirmation** lui est envoyé est envoyé au client. Ce mail est constitué d'un fichier XML **(*M2)** contenant :

- Le numéro de séjour attribué à la demande
- Le nom et le numéro de l'adhérent
- Le nom du thème, la date, l'information 'séjour validé'
- Le nom du parking, le numéro de la place et l'immatriculation de la voiture prévue
- 3 propositions d'appartement pour la nuit du samedi au dimanche et les prix de séjour correspondant (appart +2 journées de locations de la voiture)

Le client valide le séjour par un envoi de message XML **(*M3)** comprenant comme seules informations:

- Numéro de séjour, l'information 'séjour validé' et la référence de l'appartement choisi

A réception de ce message **(*P2)**, le SI de l'agence doit effectuer les opérations suivantes

- Enregistrement de la confirmation du séjour (de non confirmé à confirmé)
- Enregistrement de la réservation de la voiture pour la semaine indiquée **(*R4)**
- Envoi du SMS au contrôleur (si nécessaire)
- Rédaction **(*J4)** et envoi (hors de votre responsabilité) d'un message JSON confirmant la réservation du logement (exemple ci-joint en annexe conforme à l'API de RBNP)

A la fin de la location, le client enregistre la position de la place de parking (nom du parking et numéro de la place) où il a laissé le véhicule et attribue une note (sur 5) et un commentaire à son séjour ;

Si le séjour n'est pas réalisable, un mail d'**information** est envoyé au client. Ce mail est constitué d'un fichier XML **(*M4)** contenant :

- Le nom et le numéro de l'adhérent
- Le nom du thème, la date, l'information 'séjour impossible, veuillez choisir une autre date'

Travail demandé :

Construire la partie gestion des données (enregistrement, mise à jour, sélection, extraction mise en forme ...)
Les fonctionnalités liées à la construction d'un site web, à l'envoi et la réception de fichiers ou de message sont hors du travail demandé. Votre rôle se limitera à la mise en forme des informations (construction des messages) à envoyer.

1) La base de données

A partir des informations ci-dessus décrivant le fonctionnement de l'association Escapade, vous construirez une base de données lui permettant d'enregistrer les informations nécessaires à sa gestion courante, et de répondre aux questions suivantes. Par exemple et sans que cela soit exhaustif :

Pour les voitures :

- qui entretient une voiture donnée
- la liste des voitures et leur disponibilité
- le suivi d'entretien des voitures, quoi, quand ...
- les locations effectuées par les voitures et par qui

Pour les séjours :

- quel thème
- quand

Pour les clients :

- qui
- quel séjour
- quelle voiture loué
- quel logement

Pour les:

2) Les requêtes SQL

Vous rédigerez certaines des requêtes SQL nécessaires à l'exécution de l'activité de l'association.

1. Liste des clients (par numéro de client)
2. Saisie d'un nouveau client
3. Liste des voitures, de leur position et de leur disponibilité
4. Sélection d'une voiture disponible dans un arrondissement
5. Requête de mise à jour de la place de parking d'un véhicule identifié par son immatriculation
6. Combien d'opérations de maintenance sur une voiture identifiée par son immatriculation
7. Enregistrement du retour d'une voiture
8. Nombre de voitures contrôlées par chacun des contrôleurs
9. Liste des voitures indisponible et du motif correspondant
10. Enregistrement d'une opération de maintenance par un des contrôleurs sur une voiture identifiée par son immatriculation

3) Rédaction des messages

- (*M1)** : demande XML de séjour (depuis client)
- (*M2)** : confirmation XML du séjour (Escapade vers client)
- (*M3)** : validation du séjour (client vers Escapade)
- (*M4)** : information séjour non validé (vers le client)
- (*P1)** : requête Xpath d'analyse du message XML ***M1**
- (*P2)** : requête Xpath d'analyse du message XML ***M2**
- (*J1)** : requête JSON vers RBNP (exemple fourni en annexe)
- (*J2)** : réponse RBNP sur disponibilité logement (exemple fourni en annexe)
- (*J3)** : recherche de logement dans la requête ***J2**
- (*J4)** : réservation logement (Escapade vers RBNP)

Pour les messages XML, vous proposerez un exemple de la construction de ces messages (ou des exemples si cela vous semble pertinent) et rédigerez les DTD correspondantes.

Pour les messages JSON les exemples étant fournis, il n'y a rien à rédiger ;

Pour le ***J4**, seule une trame est fournie ; vous complèterez cette trame pour construire un exemple de message

4) Un logiciel de test pour valider vos requêtes et messages

Vous rédigerez un programme permettant de tester vos requêtes et messages

L'interface utilisateur pourra se limiter à un menu en mode console permettant d'activer les diverses fonctions à tester.

Votre interface utilisateur devra au moins tester les fonctions suivantes (pensez aux opérations intermédiaires nécessaires mais non décrites) :

1. Réception d'une demande client

- A partir d'un fichier exemple ***M1** généré par vos soins (sur la base d'une demande pour la semaine 14)
- Cette fonction procédera à la recherche d'une voiture dans le fichier XML
- Puis d'appartements (3 si possibles) dans le fichier JSON ***J2** (réponse à une requête ***J1**).
- Puis à l'écriture du mail de confirmation (fichier XML) ***M2**

2. Réception de la validation client

- A partir du fichier exemple ***M3** généré par vos soins (validation en continuation de l'exemple ***M2** précédent)
- Enregistrement de la réservation de la voiture qui passe à non disponible et dont la location est enregistrée dans le fichier adéquat.
- Rédaction du message de réservation ***J4** (au fichier JSON) conforme à la trame exemple fournie en annexe.

3. Affichage de quelques informations statistiques

- Historique d'une voiture : immatriculation, nombre de location et liste de toutes les interventions
- Historique d'un client : liste de ses séjours, date thème, véhicule loué, adresse de l'appart, notes accordées
- Rentabilité d'une voiture : cumul du montant de toutes les locations
- Thème le plus apprécié par les clients : celui choisi le plus grand nombre de fois

Les rendus :

Un rapport composé de 2 parties comprenant

Partie 1 : à rendre le jour de la séance n°5

- Le schéma de conception de votre base de données
- Le schéma relationnel de votre base de données
- Les scripts de construction des tables
- Les scripts de peuplement de votre base de données
- Les requêtes demandées (10)

Partie 2 : à rendre à la fin de la séance n°9

- Les exemples de documents XML ***M1**, ***M2**, ***M3**, ***M4** avec leur DTD
- Les requêtes XPath ***P1**, ***P2**
- La requête d'extraction ***J3**
- Un exemple de document ***J4**
- Le code C# de votre application

Ce qui vous est fourni

- Un fichier TXT comportant la liste des parkings partenaires d'Escapade
- Le fichier JSON ***J1** d'interrogation de l'API de RBNP pour Paris en semaine 14
- Un fichier exemple JSON ***J2** en réponse à une requête JSON portant sur paris/semaine 14 (la semaine 14 sera celle qui servira de base à l'exemple d'exécution prévue)
- La trame de la requête ***J3**

Vous avez effectué la première partie du projet fil rouge Escapade, et vous vous préparez à commencer la 2ème partie.

Cette deuxième partie est consacrée :

d'une part à la construction des messages d'informations entre l'application de gestion de l'agence Escapade et son environnement

(BD, serveur RBNP, application client sur le mobile des utilisateurs....)

et en complément à la construction d'un programme de présentation des fonctionnalités de l'application de gestion prévue.

Il n'est pas demandé un programme de gestion complet de l'agence Escapade mais seulement une démo démontrant vos capacités à effectuer l'informatisation complète de l'agence et à emporter ainsi le marché face à la concurrence

Vous trouverez ci-joint :

- 1) Un document ("Compléments d'informations pour le rendu de la partie 2") vous précisant les rendus demandés pour cette partie 2 (programme de démo, rapport et annexes)
- 2) Le fichier **AirBNPfinal.json** qui est la réponse à la requête simulée E4 dont vous aurez besoin pour la réalisation de votre programme de démo
- 3) Un dossier partie1 qui contient une version possible (**ce n'est PAS LA** solution de la partie1)
Ce dossier contient ce qui vous était demandé : un MCD répondant aux besoins de l'agence ESCAPADE, le script de création de table, un script de peuplement, les requêtes

Vous pouvez, mais ce n'est pas une obligation, utiliser cette base de donnée comme point de départ pour votre partie 2. Au contraire, vous êtes incités et vivement encouragés à continuer sur la base du travail que vous avez réalisé pour la partie1.

La BD fournie n'est qu'une roue de secours proposée pour permettre aux groupes ayant conçu une BD non utilisable lors de la partie 1, de continuer quand même le travail de la partie 2.

Vous pouvez demander à votre chargé de TD un avis sur le caractère "utilisable" de votre BD de la partie 1. Votre chargé de TD pourra au vu du MCD vous donner un avis (pas une correction) sur l'évaluation du type "vous pouvez continuer à utiliser votre BD" ou "il vaut mieux repartir sur une base saine", mais pas plus...

Attention : La bd fournie est fournie à titre indicatif, sans garantie d'aucune forme. Il est de votre responsabilité de la valider pour votre vision du projet, de la rectifier, de la modifier ou de la compléter si nécessaire.

D'une manière plus générale, si vous continuez avec VOTRE conception de la BD de l'agence Escapade, vous avez toute liberté pour faire évoluer votre BD à partir des réflexions que pourraient vous inspirer la BD fournie ou à partir d'un complément d'identification des besoins qui apparaîtrait lors de la réalisation de la partie 2.

Si vous êtes amenés à modifier votre BD initiale (ce qui est tout à fait autorisé et non pénalisé) vous aurez toutefois à préciser les modifications effectuées et leur motivation dans le rapport du rendu n°2 et bien sur à joindre à votre rapport la dernière version du MCD.

Date de rendu :

Le jour du TD9, mais cela peut encore évoluer

Programme fil rouge – notations

Après une première consultation rapide des documents constituant ces premiers rendus, il semble qu'une mise au point s'impose. Si certains rendus sont en accord avec ce que l'on peut attendre d'un travail de niveau bac+3, ce n'est pas le cas pour tous.

Je pense donc utile de préciser les attentes et la construction du barème de notation correspondant.

Vous devez rendre un Rapport :

- ⇒ c'est un document (au format pdf **exclusivement**) comprenant une page de garde propre comprenant l'objet du rapport ainsi que son ou ses auteurs, la date de rédaction et le contexte de rédaction (matière, sujet, nom du module etc ... à votre choix)
- ⇒ c'est un document structuré et agréable à lire comprenant :
 - Un sommaire, une introduction, une conclusion => selon les cas. Mais en aucun cas une accumulation brute de documents sans liens entre eux.
 - Vos réflexions sur le travail fourni et les résultats obtenus
 - Vos explications sur les choix et les interprétations que vous avez été amenés à faire à partir du sujet initial

Il est inadmissible (et cela sera sanctionné) de voir des rapports envoyés ne comportant même pas le nom de leurs auteurs.

Les documents envoyés doivent être pour fichiers de codes regroupés dans un fichier compressé (ZIP) et pour le rapport sous forme d'un fichier unique pdf. Les fichiers zip et pdf doivent avoir un nom permettant leur identification :

Exemple : BD_pfr_rendu1_nom1_nom2

Ce n'est pas à votre correcteur de faire un travail de recherche pour trouver à qui attribuer un rapport anonyme.

Pour ce qui en est du barème : les notations des parties 1 et 2 (sur 10 pour chacune des parties) se font sur 3 critères :

Quantitatif : la quantité de travail réalisé (environ 60%)

Qualitatif : la qualité du travail réalisé (environ 20%)

Evaluation globale du correcteur (environ 20%)

Vous comprendrez je n'en doute pas la nécessité du soin apporté à vos rendus...

Compléments d'informations pour le rendu de la partie 2

Cette partie de l'informatisation de l'agence ESCAPADE concerne la gestion de la relation client, conformément à l'énoncé initial.

Il vous est demandé une présentation du travail réalisé sous forme d'un programme de présentation des fonctionnalités du programme de gestion client.

C'est ce programme de démonstration qui sera présenté à l'agence Escapade pour évaluation de la solution proposée à leur demande d'informatisation.

Le programme consistera en la simulation des situations de gestion suivantes :

- 1) Un client réserve un séjour
- 2) Check-out d'un séjour
- 3) Edition d'un tableau de bord statistique de l'activité de l'agence.

Le programme de démonstration présentera pour chacune des étapes des situations présentées la ou les actions exécutées par votre programme.

Chacune de ces actions est séparée de la suivante par une pause et par un "appuyez sur une touche pour continuer" afin de permettre la visualisation de l'action effectuée et d'obtenir une action semi-automatique limitant l'action du spectateur à l'appuyer d'une touche pour obtenir l'avancement du déroulement de la démonstration.

Les rendus :

- 1) Le programme de démonstration et tout ce qui est nécessaire à son exécution
 - Les scripts SQL (code SQL) permettant la génération de votre base de données ; génération des tables, peuplement de tables, création d'utilisateur ...
 - Code C# de votre programme ; répertoire complet de votre projet visual studio sous forme d'un fichier compressé (zip,rar...) contenant tout votre projet.

L'accès à la BD devra se faire par un utilisateur identique pour tous les projets :

login : escapade , password : esilv.

Pensez à créer cet utilisateur lors de la génération de la BD

Les actions effectuées par l'évaluateur doivent se limiter à lancer l'exécution de vos scripts sql pour la génération de votre BD puis à lancer l'exécution de votre projet C#.

Toute autre action nécessaire de l'évaluateur pour visualiser votre programme de démonstration rendra votre programme de démonstration NON EVALUABLE.

A vous de vérifier que tout est correct avant de poster votre travail sur Moodle

- 2) Un rapport contenant à minima
 - Une présentation de votre travail
 - Le MCD de la BD
 - Les choix de gestion que vous avez pu être amené à effectuer
- 3) en annexe:
 - les éditions des messages émis ou reçus (M1, M2, M3)
 - les requêtes XPath utilisées
 - les requêtes SQL utilisées et une copie écran des résultats obtenus
- 4) Tout autre document que vous considérerez comme pertinent

Rappel :

Pensez que c'est sur la lecture de ce rapport et sur l'exécution du programme de démonstration que se fondera l'évaluation de l'agence Escapade et sa motivation à nous passer la commande de son informatisation (et donc votre évaluation notée)

Description des situations du programme de démonstration et des étapes correspondantes (E1) à ...(Ex).

1) Un client réserve un séjour :

(E1) Le message XML (*M1) émis par l'application cliente (sur le portable du client) contenant : nom du client, séjour souhaité, date.

Client : M. votrenom

Adresse : ESILV, la défense

Date : semaine 14

Séjour : visite de la Défense

(pour information et pour cet exercice, la Défense est rattachée au 16^{ème} arrondissement)

(E2) Le programme vérifie si le client existe

S'il est nouveau création d'un nouveau client

S'il existe déjà : récupération de son numéro client

(E3) Le programme sélectionne une voiture

Dans un parking du 16 s'il y a en a une (et dans la démo, il n'y en aura pas)

Sinon : il déplace une voiture disponible dans un autre arrondissement (changement de parking) et la réserve.

(E4) (simulée.... Rien à faire) Le programme émet une requête RBNP pour obtenir la situation des logements à Paris pour la semaine demandée (la 14).

Et une réponse vous est renvoyée par RBNP (le fichier vous est fourni sur Moodle).

(E5) Le programme sélectionne un appartement dans le fichier JSON reçu répondant aux caractéristiques suivantes :

Arrondissement : 16^{ème}

Nombre de chambres : 1

Evaluation minimum : 4,5

Et bien sur disponible

(E6) Le programme effectue la réservation (en mode non confirmé) du séjour et envoie au client le message XML de validation du séjour (message *M2 de l'énoncé) avec le prix de l'appartement sélectionné.

(E7) Validation du séjour par le client

Emission du message XML (*M3) par le client

Et passage de la réservation en mode confirmé

2) Fin de séjour : Check-out Client et opération de remise en service de la voiture

- Enregistrement de la position de la voiture rendue par le client
- Contrôle de la voiture par un des vérificateurs
- Le vérificateur constate que la voiture doit être nettoyée => indisponible
- Nettoyage du véhicule
- Remise en service de la voiture

3) Tableau de bord Escapade et quelques statistiques

- Historique des interventions sur un véhicule (celui utilisé par le client précédent par exemple)
- Historique d'un client (même remarque)
- Rentabilité d'un véhicule ; nombre de locations, etc ... informations à choisir par vos soins... (même remarque)
- Et tout autre tableau vous semblant utile et pertinent

Lire attentivement ce document pour les nouvelles instructions de rendu (2^{ème} rendu) de votre programme fil rouge.

Bonjour a tous,

Préambule

Il était précédemment prévu que vous déposiez sur Moddle votre script de réinstallation de votre Bd ainsi que votre code c#. Ce qui imposait de réinstaller votre BD pour tester votre code.

Toute réinstallation est susceptible de poser des problèmes. Et si la réinstallation de votre BD pose un quelconque problème (nom d'utilisateur, mot de passe, version de Sql différente, machine différente etc..) votre code c# ne marchera pas et votre note s'en ressentira.

Afin d'éviter tout problème lié à une réinstallation de votre BD, j'ai installé un serveur MYSQL auquel vous avez accès. Un compte a été créé pour chaque étudiant ainsi qu'une base de données associée à ce compte. Sur cette database (et sur cette database uniquement), chaque étudiant dispose de tous les droits lui permettant de travailler comme s'il était en local.

Rendu Fil Rouge (2^{ème})

Le 2^{ème} rendu Fil rouge s'effectuera donc ainsi :

- Installation par vos soins de votre base de données sur le serveur commun
- Contrôle du fonctionnement de votre code avec connexion sur le serveur commun
- Dépôt sur Moodle de votre Code cs et de votre rapport (au format pdf)

La correction comportera une phase d'exécution de votre programme de démonstration qui devra s'exécuter en se connectant au serveur commun sous votre nom.

Utilisation de la base de données commune

Très peu de changements pour vous si vous suivez attentivement les instructions ci-dessous :

1) avec MySQL Workbench

Ouvrir MySQL WorkBench

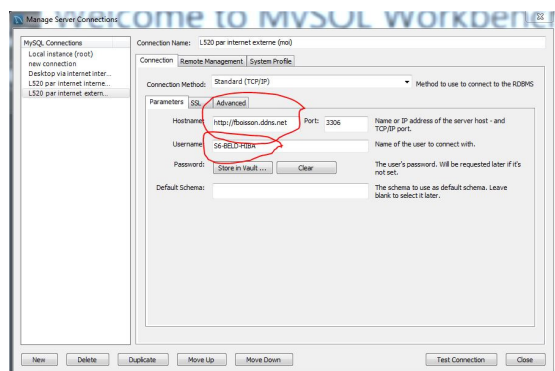
Créer une nouvelle connexion avec les paramètres suivants

hostName : <http://fboisson.ddns.net> au lieu localhost

Le Port ne change pas : 3306

Le Username vous est imposé : vous le trouverez dans le document joint login-BD.pdf

Le Password qui vous est affecté est votre numéro d'étudiant



En utilisant ces paramètres vous disposerez d'une connexion vers la base de données commune. Vous y trouverez déjà créée, la base de données qui vous est affectée.

Il vous reste à sélectionner cette base de donnée (use DataBaseName ou doubleClick sur le nom de la BD) pour pouvoir créer vos tables, les peupler et exécuter vos requêtes.

2) Avec C#

La seule modification de code C# à faire par vous concerne la ligne définissant la connexion à la base de donnée (Cette ligne est la ligne MySqlConnection =).

```
string connectionString = "SERVER=localhost;PORT=3306;DATABASE=loueur;UID=root;PASSWORD=rumtot;";  
MySqlConnection connection = new MySqlConnection(connectionString);
```

Il vous suffit donc de remplacer les paramètres de connexion à votre serveur local par les paramètres de connexion à mon serveur:

- SERVER = localhost devient SERVER = fboisson.ddns.net
- PORT = 3306 est inchangé
- DATABASE = votre nom de database devient pour vous DATABASE = votre nom de database à lire dans le document ci-joint
- UID = votre nom de login (en general root) devient pour vous UID = votre nom de login à lire dans le document ci-joint (commençant par S6-)
- PASSWORD = votre numéro d'étudiant (voir vos notes)

Attention aux - et aux _

3) Précautions d'usage

Le serveur étant un PC sous Windows normal (pas particulièrement puissant) et la liaison une liaison (fibre) privée, si vous vous connectez et travaillez ensemble en ligne à 250 les performances du système ne sont pas garanties.

Donc vous êtes invités à mettre au point votre programme en local et une fois que tout fonctionne à transférer la partie BD sur le serveur commun, à vérifier que tout fonctionne et à ne procéder en ligne qu'aux ajustements nécessaires.

De même évitez de vous y prendre à la dernière minute pour transférer vos Bd pour éviter les embouteillages de dernière minute...

Gestion du rendu

- Vous réalisez votre programme de démonstration en local,
- Vous l'essayez en local
- Vous testez avec votre BD Installée sur le serveur commun
- Vous essayez votre programme C# en considérant que les tests du correcteur seront faits sur un PC fonctionnant sous Windows
- Après avoir tout vérifié, vous déposez sur Moodle votre programme C#
- Il doit pouvoir être rechargé et utilisé tel quel par le chargé de TD qui effectuera la correction de votre programme fil rouge (cela doit être sans problème puisque le programme C# contient vos paramètres de connexion au serveur commun et qu'il a été testé depuis votre machine)

Et ainsi pas de problèmes liés à la réinstallation de votre BD.

Les documents composant le 2^{ème} rendu programme fil rouge.

Bonjour à tous,

Préambule

Le principe du rendu est le suivant : vous réalisez et présentez le programme installé au siège de la société Escapade. Les objectifs et fonctionnalités de ce logiciel ont été décrits dans les précédents énoncés.

Votre travail et la démonstration correspondante se limite à la réalisation du logiciel coté société Escapade (ce qui inclus la gestion de sa base de données SQL) et à la définition des messages échangés avec les logiciels extérieurs.

La réalisation du logiciel s'exécutant sur les mobiles coté client de la société Escapade est ainsi hors de votre périmètre de travail.

Seuls sont à votre charge la définition et la construction des messages échangés entre le logiciel d'Escapade et le logiciel fonctionnant sur les téléphones de ses clients

Des documents successifs vous ont été fournis sur Moodle. Des questions ont été posées.

Des éclaircissements s'imposent.

Et je vais en profiter pour remettre dans l'ordre l'ensemble des documents qui vous ont été envoyés sur Moodle

Et une demande complémentaire (oubliée sur les énoncés précédents) :

Dans l'étape (E5), il manque un point important : informer RBNP que l'appartement choisi est maintenant réservé. Donc génération d'un message Json (***Jx**), sous forme d'un fichier Json, contenant les informations suivantes pour l'appartement choisi :

host_id, room_id, week et availability avec pour valeur no bien sur.

Le 2^{ème} rendu

Le principe du rendu est de présenter un programme de présentation des fonctionnalités du programme que vous avez pour assurer la gestion de l'activité de la société Escapade.

Le programme de gestion réalisé n'est pas complet, il manque des fonctionnalités, les fonctionnalités codées sont restreintes. Certes mais il sera suffisant pour convaincre Escapade de votre capacité à réaliser le logiciel complet.

Le 2^{ème} rendu est donc construit autour de cette idée : le chargé de Tds (jouant le rôle du représentant de la société Escapade) doit être convaincu et passer la commande finale (.....votre note).

Le 2^{ème} rendu Fil rouge comportera donc :

- Le programme de démonstration des fonctionnalités demandées dans un document précédent, le **document XXX** (le fichier de code .cs uniquement ... il sera rechargé dans un projet visual studio préparé à cet effet)
- La base de données nécessaire au fonctionnement de ce programme installée par vos soins sur le serveur fboisson.dds.net
Attention : fboisson.dds.net et pas http://fboisson.dds.net comme indiqué par erreur dans un des documents précédents.
Votre logiciel de démonstration doit être essayé par vos soins et fonctionner avec la BD installée sur le serveur fboisson.dds.net
- Un rapport (au format pdf) décrivant votre travail (cf le **document XXX** décrivant ce que doit contenir à minima un rapport de travail). Tout document qui vous paraîtrait pertinent
- Les fichiers XML créés (*M1), (*M2), (*M3),

Les documents XML :

Les documents XML générés durant les étapes des fonctionnalités demandés feront donc faire l'objet d'un rendu sous forme de fichiers séparés et/ou, à vous de juger, d'insertion complète ou sous forme d'extraits, en annexe dans le rapport rendu.

Les étapes du programme de démonstration et les messages correspondants

1) Check-in

- (E1) le client souhaite réserver un séjour
 - Le message (*M1) émis par l'application du mobile
 - Vous générez donc le message à la main => fichier XML
 - Votre programme de démonstration l'affiche et le message est joint à votre rapport en annexe
- (E2) le programme vérifie si le client existe
 - Parsing (lecture) du fichier reçu (*M1) ou analyse du fichier par une requête Xpath (*P1) pour y trouver le nom du client
 - Votre code doit être capable de traiter les deux cas : existe et n'existe pas
 - Requête SQL, requête Xpath (*P1) et extrait de code dans le rapport
 - Pour le programme de démo choisir l'une ou l'autre des situations
- (E3) votre programme sélectionne une voiture
 - Requête dans le rapport
 - Visualisation du résultat de la requête dans le programme de démo
- (E4) Le programme émet une requête simulée vers RBNP et reçoit le résultat
 - Cette étape est simulée donc
 - Rien si ce n'est peut être un message informant que cela est fait
- (E5) Sélection d'un appartement
 - Dans le rapport : comment faites-vous ?
 - Extrait de code, explications de la méthode employée : Parsing de (*J2), JsonXpath...
 - Génération du message de confirmation (*Jx) à destination de RBNB
 - Dans le rapport message (*Jx) généré et extrait de code correspondant
 - Dans la démo : Affichage du résultat : appartement choisi et message (*Jx)
- (E6) Le programme effectue la réservation (en mode non confirmé), génère et envoie le message (*M2)
 - L'envoi du message consiste à placer le message (*M2) dans un fichier XML
 - Dans le rapport les requêtes, le message généré et l'extrait de code qui génère le message
 - Et dans un fichier, le fichier XML généré
 - Dans la démo : affichage du résultat des requêtes et du message envoyé.
- (E7) Validation du séjour. Le client envoie le message (*M3)
 - (*M3) est envoyé par le portable du client
 - Vous construisez donc le message (*M3) à la main => fichier XML
 - Puis par votre programme de gestion analyse du fichier (*M3) par une ou des requêtes Xpath (*P2) pour en extraire les informations nécessaires au passage en mode confirmé du séjour dans le fichier SQL de l'agence.
 - Donc dans le rapport : le message XML généré, les requêtes Xpath utilisées pour son analyse, la ou les requêtes SQL et extrait de code si nécessaire.
 - Le programme de démo affiche ce qu'il fait au fur et à mesure

On ne traitera pas le cas où la demande de séjour du client n'est pas réalisable (donc la génération du message (*M4) n'est pas demandée.

2) Check-out

- Même principe mais il n'y a qu'une étape
- Le programme de démo affiche la succession des opérations et on retrouve dans le rapport la succession des requêtes utilisées

3) Tableau de bord

- Idem : trois affichages successifs et trois requêtes (lisibles) avec leurs résultats dans le rapport

Les documents composant le 2^{ème} rendu programme fil rouge.

Bonjour à tous,

Préambule

Le principe du rendu est le suivant : vous réalisez et présentez le programme installé au siège de la société Escapade. Les objectifs et fonctionnalités de ce logiciel ont été décrits dans les précédents énoncés.

Votre travail et la démonstration correspondante se limite à la réalisation du logiciel coté société Escapade (ce qui inclut la gestion de sa base de données SQL) et à la définition des messages échangés avec les logiciels extérieurs.

La réalisation du logiciel s'exécutant sur les mobiles coté client de la société Escapade est ainsi hors de votre périmètre de travail.

Seuls sont à votre charge la définition et la construction des messages échangés entre le logiciel d'Escapade et le logiciel fonctionnant sur les téléphones de ses clients

Des documents successifs vous ont été fournis sur Moodle. Des questions ont été posées.

Des éclaircissements s'imposent.

Et je vais en profiter pour remettre dans l'ordre l'ensemble des documents qui vous ont été envoyés sur Moodle

Et une demande complémentaire (oubliée sur les énoncés précédents) :

Dans l'étape (E5), il manque un point important : informer RBNP que l'appartement choisi est maintenant réservé. Donc génération d'un message Json (***Jx**), sous forme d'un fichier Json, contenant les informations suivantes pour l'appartement choisi :

host_id, room_id, week et availability avec pour valeur no bien sur.

Le 2^{ème} rendu

Le principe du rendu est de présenter un programme de présentation des fonctionnalités du programme que vous avez pour assurer la gestion de l'activité de la société Escapade.

Le programme de gestion réalisé n'est pas complet, il manque des fonctionnalités, les fonctionnalités codées sont restreintes. Certes mais il sera suffisant pour convaincre Escapade de votre capacité à réaliser le logiciel complet.

Le 2^{ème} rendu est donc construit autour de cette idée : le chargé de Tds (jouant le rôle du représentant de la société Escapade) doit être convaincu et passer la commande finale (.....votre note).

Le 2^{ème} rendu Fil rouge comportera donc :

- Le programme de démonstration des fonctionnalités demandées dans un document précédent, le **document XXX** (le fichier de code .cs uniquement ... il sera rechargé dans un projet visual studio préparé à cet effet)
- La base de données nécessaire au fonctionnement de ce programme installée par vos soins sur le serveur fboisson.dds.net
Attention : fboisson.dds.net et pas http://fboisson.dds.net comme indiqué par erreur dans un des documents précédents.
Votre logiciel de démonstration doit être essayé par vos soins et fonctionner avec la BD installée sur le serveur fboisson.dds.net
- Un rapport (au format pdf) décrivant votre travail (cf le **document XXX** décrivant ce que doit contenir à minima un rapport de travail). Tout document qui vous paraîtrait pertinent
- Les fichiers XML créés (*M1), (*M2), (*M3),

Les documents XML :

Les documents XML générés durant les étapes des fonctionnalités demandés feront donc faire l'objet d'un rendu sous forme de fichiers séparés et/ou, à vous de juger, d'insertion complète ou sous forme d'extraits, en annexe dans le rapport rendu.

Les étapes du programme de démonstration et les messages correspondants

1) Check-in

- (E1) le client souhaite réserver un séjour
 - Le message (*M1) émis par l'application du mobile
 - Vous générez donc le message à la main => fichier XML
 - Votre programme de démonstration l'affiche et le message est joint à votre rapport en annexe
- (E2) le programme vérifie si le client existe
 - Parsing (lecture) du fichier reçu (*M1) ou analyse du fichier par une requête Xpath (*P1) pour y trouver le nom du client
 - Votre code doit être capable de traiter les deux cas : existe et n'existe pas
 - Requête SQL, requête Xpath (*P1) et extrait de code dans le rapport
 - Pour le programme de démo choisir l'une ou l'autre des situations
- (E3) votre programme sélectionne une voiture
 - Requête dans le rapport
 - Visualisation du résultat de la requête dans le programme de démo
- (E4) Le programme émet une requête simulée vers RBNP et reçoit le résultat
 - Cette étape est simulée donc
 - Rien si ce n'est peut être un message informant que cela est fait
- (E5) Sélection d'un appartement
 - Dans le rapport : comment faites-vous ?
 - Extrait de code, explications de la méthode employée : Parsing de (*J2), JsonXpath...
 - Génération du message de confirmation (*Jx) à destination de RBNB
 - Dans le rapport message (*Jx) généré et extrait de code correspondant
 - Dans la démo : Affichage du résultat : appartement choisi et message (*Jx)
- (E6) Le programme effectue la réservation (en mode non confirmé), génère et envoie le message (*M2)
 - L'envoi du message consiste à placer le message (*M2) dans un fichier XML
 - Dans le rapport les requêtes, le message généré et l'extrait de code qui génère le message
 - Et dans un fichier, le fichier XML généré
 - Dans la démo : affichage du résultat des requêtes et du message envoyé.
- (E7) Validation du séjour. Le client envoie le message (*M3)
 - (*M3) est envoyé par le portable du client
 - Vous construisez donc le message (*M3) à la main => fichier XML
 - Puis par votre programme de gestion analyse du fichier (*M3) par une ou des requêtes Xpath (*P2) pour en extraire les informations nécessaires au passage en mode confirmé du séjour dans le fichier SQL de l'agence.
 - Donc dans le rapport : le message XML généré, les requêtes Xpath utilisées pour son analyse, la ou les requêtes SQL et extrait de code si nécessaire.
 - Le programme de démo affiche ce qu'il fait au fur et à mesure

On ne traitera pas le cas où la demande de séjour du client n'est pas réalisable (donc la génération du message (*M4) n'est pas demandée.

2) Check-out

- Même principe mais il n'y a qu'une étape
- Le programme de démo affiche la succession des opérations et on retrouve dans le rapport la succession des requêtes utilisées

3) Tableau de bord

- Idem : trois affichages successifs et trois requêtes (lisibles) avec leurs résultats dans le rapport