

ETUDE DE CAS

Durée : 4 heures

Coefficient : 5

Cas Dane

Ce sujet comporte 18 pages dont 12 pages de documentation.

La candidate ou le candidat doit vérifier que le sujet qui lui est remis est complet.

Aucun matériel ni document autorisé

Dossier documentaire :

Document 1	Description de la base de données <i>db_us_num</i>	page 7
Document 2	Contenus des tables (extraits)	page 7
Document 3	Extrait du tableau de bord du Lycée Vanille Bourbon Année scolaire 2015-2016	page 8
Document 4	Type de graphique attendu.....	page 9
Document 5	Schéma de principe de la génération des graphiques	page 9
Document 6	Note de M. Vans.....	page 10
Document 7	Extraits du code dédié à la génération des graphiques	page 11
Document 8	Diagramme des classes métiers (extraits)	page 13
Document 9	Description partielle des classes métiers utilisées dans le document 8	page 13
Document 10	Classe technique ArrayObject (extraits de la documentation)	page 16
Document 11	Formulaire papier proposé par la Dane à remplir lors d'une observation	page 17
Document 12	Extrait du schéma conceptuel existant fourni dans les deux représentations les plus courantes.....	page 17
Document 13	Proposition de maquette	page 18

Barème

Mission 1	Nouvelles statistiques	10 points
Mission 2	Ajout d'un graphique au tableau de bord	30 points
Mission 3	Synchronisation de la base de données	30 points
Mission 4	Évolution de la structure de la base de données pour <i>Usnum</i>	30 points
	Total	100 points

Présentation du contexte

La **Délégation Académique au Numérique pour l'Éducation (Dane)** de l'académie de R. a été créée à la rentrée 2014 dans les locaux du rectorat. Madame Milos a été nommée déléguée académique au numérique, conseillère technique du recteur.

La **Dane** a pour vocation de promouvoir les usages du numérique pédagogique dans les écoles, les collèges et les lycées de l'académie.

Son champ d'action est défini en quatre pôles :

1. Suivi des usages ;
2. Formation au et par le numérique ;
3. Enquêtes et outils de pilotage ;
4. Information et communication.

Pour mener à bien ses missions, l'équipe de la **Dane** est composée, outre de Madame Milos et d'un groupe de formateurs, d'une secrétaire et de professeurs "chargés de mission" mis à disposition par le recteur. Un des chargés de mission, M. Santes a en charge les pôles un et trois. C'est le troisième pôle, "enquêtes et outils de pilotage", qui doit être développé.

Pour faire le lien entre la **Dane** et les établissements scolaires du secondaire (collèges et lycées), le rectorat a nommé des référents ; un référent est une professeure ou un professeur chargé.e de suivre un ou plusieurs établissements.

Le pôle "Enquêtes et outils de pilotage" est limité pour l'instant aux statistiques d'usages de l'**Environnement Numérique de Travail (ENT)**. À chaque fois qu'un utilisateur (élève, étudiant, professeur, parent ou personnel administratif) se connecte à l'**ENT**, des compteurs d'usages s'incrémentent en fonction des applications et des services activés. Ces informations alimentent tous les jours une base de données à partir de laquelle est généré un tableau de bord. Les référents disposent ainsi, pour chaque établissement, de plusieurs graphiques permettant de mesurer l'évolution des usages du numérique éducatif.

La **Dane** souhaite dans un premier temps faire évoluer l'existant en améliorant les tableaux de bord ainsi que la synchronisation quotidienne de la base.

Dans un deuxième temps, elle désire intégrer à la base de données existante, les usages pédagogiques du numérique éducatif hors **ENT** (applications et matériels utilisés par les enseignants). À cet effet, la **Dane** demandera aux référents de se rapprocher des établissements pour recenser les pratiques pédagogiques utilisant le numérique.

La **Dane** a fait appel à la **Division des Systèmes d'Information (DSI)** du rectorat de l'académie de R. pour répondre à ces nouveaux besoins. La **DSI** est en charge notamment de l'informatisation et du suivi des services numériques administratifs, du rectorat et des établissements scolaires. Son personnel est composé de techniciens et ingénieurs en informatique spécialisés dans le domaine des réseaux, du développement et de la sécurité.

Un ingénieur de la **DSI**, M. Vans, a été nommé pour conduire ce projet de système d'information avec M. Santes comme interlocuteur à la **Dane**.

Il a été décidé que l'application existante **Usnum** subirait une maintenance évolutive et que de nouvelles fonctionnalités seraient développées.

Vous devez intervenir dans l'équipe de M. Vans pour participer à ces développements.

			Page : 2/18

Documents à utiliser : 1, 2

Les usages observés relèvent de catégories. Pour chaque catégorie d'usages, on pourra recenser plusieurs applications. Dans la catégorie "Vie scolaire" on trouvera par exemple un logiciel de saisie des notes et bulletins, un autre dédié au cahier de textes, etc. Ces applications sont activées ou non pour chaque type d'utilisateur d'un établissement. L'application "Messagerie académique" est par exemple activée uniquement pour les enseignants et le personnel administratif dans **I'ENT** de chaque établissement.

Pour mesurer l'évolution des usages liés aux services proposés par **I'ENT**, une base de données est actualisée quotidiennement à partir des mesures d'audience qui sont les statistiques fournies par la **DSI** sur les connexions à **I'ENT**.

Les usages relevés concernent le nombre de visites global à **I'ENT** d'un établissement qu'il soit collège ou lycée, ainsi que le nombre de connexions par catégorie d'usages. Ces indicateurs, archivés sur une période d'un mois, sont stockés dans une base de données MySQL nommée *db_us_num*.

M. Santes, chargé de mission à la **Dane** a exprimé le besoin d'accéder à de nouvelles statistiques à partir de cette base :

1. Liste des noms des applications répertoriées dans la catégorie d'usages "Parcours pédagogique" ;
2. Afficher pour l'établissement 9940001A (lycée Vanille Bourbon), le minimum, le maximum et la moyenne du nombre de connexions pour l'année scolaire 2015-2016 (du 1^{er} septembre 2015 au 5 juillet 2016) ;
3. Liste des établissements (libellé et type) n'ayant aucune application répertoriée dans la catégorie d'usages "Travail collaboratif".

Votre mission**1.1 - Écrire les requêtes SQL qui répondent aux besoins exprimés par la Dane.**

			Page : 3/18

Mission 2 – Ajout d'un graphique au tableau de bord

Documents à utiliser : 1, 2, 3, 4, 5, 6, 7

IMPORTANT : la candidate ou le candidat peut choisir de présenter les éléments de code à l'aide du langage de programmation de son choix ou de pseudo-code algorithmique.

L'application actuelle offre un tableau de bord qui contient un ensemble de graphiques permettant de suivre, par établissement, l'évolution des usages au sein de **I'ENT**. Ces graphiques permettent, par exemple, d'obtenir le nombre total de connexions, les nombres de connexions par catégorie d'usages, etc.

Le document 3 donne un extrait du tableau de bord d'un établissement pour 2015-2016. Deux exemples de graphique sont présentés : l'évolution du nombre de connexions mensuelles à **I'ENT** et la répartition des visites en fonction des usages. Dans le deuxième graphique, on constate que les usages de type "Vie scolaire" (consultation des notes, appels, ...) représentent la majeure partie des visites.

Lors de la dernière réunion de travail du chargé de mission M. Santes avec les référents il a été suggéré, pour une meilleure visibilité des autres types d'usages, d'ajouter un graphique dans lequel les usages de type "Vie scolaire" ne figureraient pas (**document 4**).

Cette proposition a été validée par la déléguée académique au numérique éducatif, Mme Milos. Vous êtes chargé.e par M. Vans de modifier l'application existante afin d'intégrer ce nouveau graphique dans le tableau de bord. M. Vans vous a adressé une note détaillant le travail à effectuer.

Cette application *web* développée en *PHP* utilise la base de données *MySQL db_us_num* et exploite les technologies suivantes côté client :

- *jQuery* : bibliothèque *JavaScript* libre permettant de mettre à jour facilement le contenu d'une page *HTML* en exploitant le modèle objet de document (*DOM*) et la technologie *JavaScript* asynchrone et *XML (Ajax)* ;
- *Highcharts* : bibliothèque *JavaScript* dédiée à la création de graphiques de toutes sortes et utilisable gratuitement dans le cadre d'un usage non commercial (www.highcharts.com).

Votre mission

2.1- Expliquer l'intérêt de ce nouveau graphique pour le tableau de bord.

2.2- Indiquer sur votre copie, les lignes de codes nécessaires pour prendre en compte la note rédigée par M. Vans.

Les scripts à modifier se trouvent dans le document 7.

2.3- Proposer une solution à M. Vans afin d'éviter d'écrire la fonction `cnx_usages_horsVS` (sans mettre en œuvre la solution).

			Page : 4/18

Mission 3 – Synchronisation de la base de données

Documents à utiliser : 8, 9, 10

IMPORTANT : la candidate ou le candidat peut choisir de présenter les éléments de code à l'aide du langage de programmation de son choix ou de pseudo-code algorithmique.

La base de données *db_us_num*, présentée dans la mission 1, est actualisée quotidiennement à partir de mesures d'audience fournies par la **DSI**.

Chaque nuit un script génère deux fichiers de valeurs séparées par des points-virgules (format CSV) et les dépose sur le serveur qui héberge la base de données et l'application du tableau de bord :

- Le fichier **connexions.csv** détaille par établissement le nombre de connexions par catégorie d'usages.
- Le fichier **applications.csv** détaille par établissement et par type d'utilisateur la liste des applications activées dans **I'ENT**.

Vient ensuite une phase de synchronisation qui consiste à actualiser la base de données *db_us_num* à partir des informations contenues dans ces fichiers CSV. Ces deux sources de données (base de données *db_us_num* et fichiers CSV) sont respectivement gérées par les classes **AudiencesBdd** et **AudiencesCsv**. Afin de simplifier la synchronisation, une instance de chaque classe charge les données en mémoire sous la forme de deux dictionnaires (*lesConnexions* et *lesApplications*). La classe **Synchronisation** a la responsabilité d'actualiser la base de données à partir des informations présentes dans les fichiers CSV.

Ainsi, les nombres de connexions relevés dans le fichier **connexions.csv** sont reportés dans la base de données et lorsqu'une nouvelle application apparaît dans le fichier **applications.csv**, elle est automatiquement insérée dans la table dédiée.

Si la création d'un nouvel établissement n'est pas un événement courant, on souhaiterait cependant que cet événement soit détecté de manière automatique. Cela nécessite donc de repérer les nouveaux établissements dans les fichiers CSV puis de mettre à jour la base de données. La présence d'un nouvel établissement sera détectée par un nombre d'établissements plus important dans la source de données des fichiers CSV. Pour information la suppression d'un établissement, événement rarissime, sort de la présente étude.

Vous êtes chargé.e d'implémenter et de tester cette nouvelle fonctionnalité.

Votre mission

3.1 - Écrire la méthode *getNbEtablissements()* de la classe *Audiences*, qui retourne le nombre d'établissements chargés dans les dictionnaires (la méthode pourra se baser indifféremment sur le dictionnaire *\$lesConnexions* ou *\$lesApplications*).

3.2 - Écrire la méthode *syncEtablissements()* de la classe *Synchronisation*, qui se charge de déceler les nouveaux établissements qui doivent être ajoutés en comparant les deux sources de données. En présence d'un nouvel établissement, la mise à jour de la base se fera à l'aide de la méthode *AudiencesBdd::insertEtablissement()*.

Les tests unitaires de ces deux précédentes méthodes ont été passés avec succès. Lors des tests d'intégration, réalisés à l'aide de jeux d'essai contenant un nouvel établissement, le message ci-après apparaît lors de la synchronisation :
"Synchronisation::syncConnexions() : #1452 - Cannot add or update a child row: a foreign key constraint fails".

3.3 - Expliquer l'origine de cette erreur et donner le correctif à apporter au code.

			Page : 5/18

Mission 4 – Évolution de la structure de la base de données pour *Usnum*

Documents à utiliser : 11, 12, 13

IMPORTANT : la candidate ou le candidat présentera les évolutions de la structure de la base de données en adoptant le formalisme de son choix (schéma entité-association, diagramme de classes, ou encore schéma relationnel).

Réflexion préalable

La **Dane** souhaiterait étendre son enquête à l'utilisation, durant des séquences pédagogiques :

- d'applications hors **ENT**, comme *Google Drive*, *YouTube*, *Facebook*, etc. ;
- de matériels comme les téléphones intelligents (*smartphones*), les tablettes, les ordinateurs, etc.

Pour cela, la nouvelle fonctionnalité de l'application **Usnum**, qui sera intégrée à l'infrastructure existante du rectorat, permettra de collecter en ligne les informations issues des observations des référents.

La **Dane**, ayant un besoin urgent de ces nouvelles statistiques et sachant que le développement de la nouvelle fonctionnalité de l'application **Usnum** va prendre du temps, a dû recourir à une solution temporaire. M. Santes a demandé aux référents de remplir un formulaire papier (**document 11**) pour chacune des observations qu'il réalise dans le ou les établissements dont il a la charge. Ce formulaire sera transmis à la secrétaire de la **Dane** qui saisira les données dans une feuille de calcul de tableur. Pour rappel, un référent est une professeure ou un professeur nommé.e par le rectorat pour suivre un ou plusieurs établissements du secondaire (collèges et lycées).

M. Vans, l'ingénieur DSI, sollicité par M. Santes, vous demande dans un premier temps de réfléchir sur les conséquences de l'arrivée de cette application pour les futurs utilisateurs mais également sur la sécurité qu'il faudra mettre en place lors du développement.

Votre mission

4.1- Expliquer les inconvénients de la gestion papier des enquêtes tant pour M. Santes et la secrétaire de la Dane, que pour les référents.

4.2- Expliquer en quoi l'utilisation de la nouvelle fonctionnalité en ligne va changer les habitudes des référents et celles de la Dane. Préconiser les solutions permettant de gérer ce changement.

4.3- Donner des recommandations sur l'environnement de développement à mettre en place, à la fois sur les plans logiciels et matériels, pour la programmation de la nouvelle fonctionnalité.

Évolution de la structure de la base de données

Après analyse rapide du formulaire papier (**document 11**), M. Vans a réalisé une première maquette de saisie des données d'observation (**document 13**). Il vous demande de poursuivre le travail.

Votre mission

4.4- Proposer pour chacune des zones 1, 4, 8 et 10 le composant graphique adapté (zone de saisie, liste déroulante, etc.) afin de limiter les erreurs de saisie.

4.5- A partir de la maquette et du schéma conceptuel existant (document 12**), proposer une modélisation de la base de données qui prenne en compte les nouveaux besoins exprimés.**

			Page : 6/18

Document 1 : Description de la base de données *db_us_num***CategorieUsage** (id, libelle)

clé primaire : id

TypeUtilisateur (id, libelle)

clé primaire : id

ApplicationENT (id, nom, dateInstallation, version, idCategorieUsage)

clé primaire : id

clé étrangère : idCategorieUsage en référence à id de CategorieUsage

Etablissement (rne, libelle, type)

clé primaire : rne

// répertoire national des établissements**AppEtabActivees** (idApplicationENT, idTypeUtilisateur, rneEtablissement)

clé primaire : idApplicationENT, idTypeUtilisateur, rneEtablissement

clé étrangère : idApplicationENT en référence à id de Application

clé étrangère : idTypeUtilisateur en référence à id de TypeUtilisateur

clé étrangère : rneEtablissement en référence à rne de Etablissement

//répertorie les applications activées par type d'utilisateur et par établissement

HistoUsageEtablissement (idCategorieUsage,rneEtablissement,dateObservation,nbCnx)

clé primaire : idCategorieUsage,rneEtablissement,dateObservation

clé étrangère : idCategorieUsage en référence à id de CategorieUsage

clé étrangère : rneEtablissement en référence à rne de Etablissement

//contient le cumul mensuel (observé le dernier jour du mois) des connexions aux applications utilisées par catégorie d'usages et pour chaque établissement

HistoCnxEtablissement (rneEtablissement, dateObservation, nbCnx)

clé primaire : rneEtablissement,dateObservation

clé étrangère : rneEtablissement en référence à rne de Etablissement

// contient le cumul mensuel (observé le dernier jour du mois) des accès à l'ENT par établissement.

Document 2 : Contenus des tables (extraits)**Table CategorieUsage**

id	libelle
101	Travail collaboratif
102	Courrier électronique
108	Vie scolaire
...	

Table TypeUtilisateur

id	libelle
1	Elève
2	Enseignant
...	

Table ApplicationENT

id	nom	dateInstallation	version	idCategorieUsage
26	Webmail	2008/10/10	2	102
27	Moodle	2009/07/15	3	101
...				

			Page : 7/18

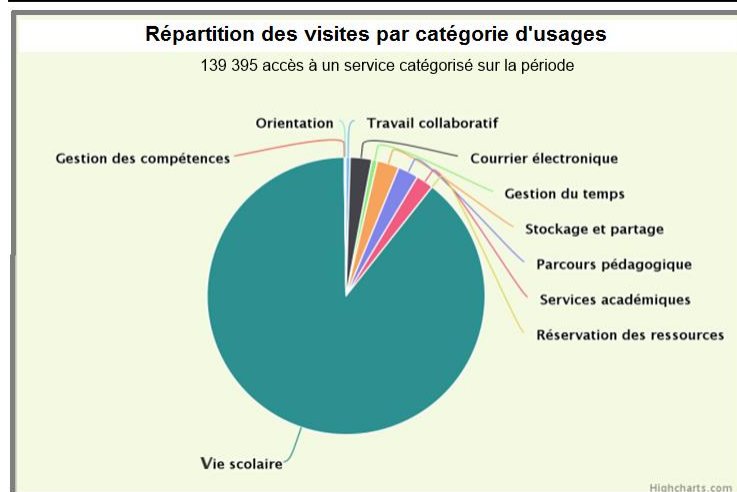
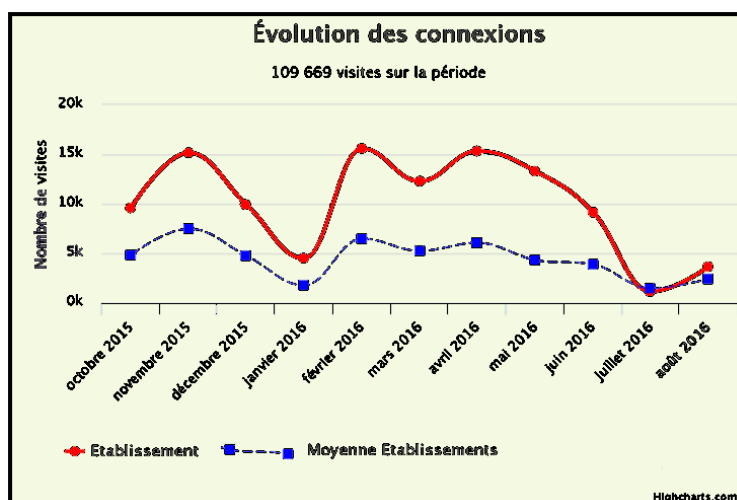
Table Etablissement		
rne	libelle	type
9940001A	Lycée Vanille Bourbon	LYC
9940001B	Collège Le Busard	CLG
...		

Table AppEtabActivees		
idApplicationENT	idTypeUtilisateur	rneEtablissement
26	2	9940001A
27	2	9940001B
27	1	9940001B
...		

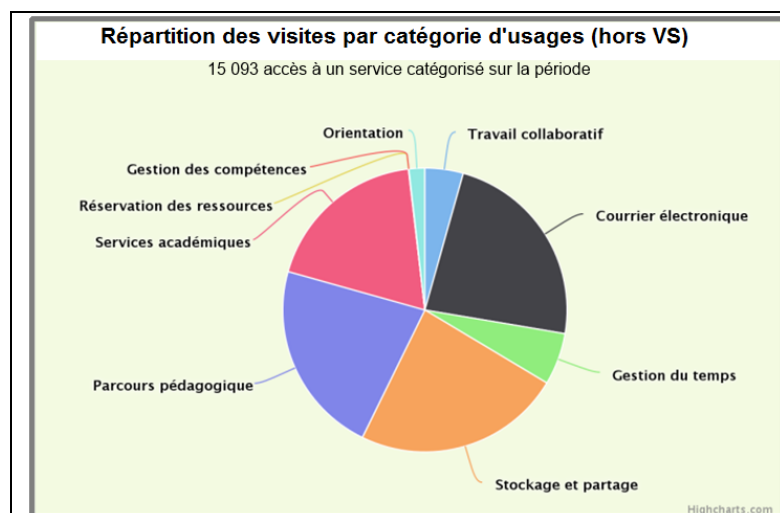
Table HistoUsageEtablissement			
idCategorieUsage	rneEtablissement	dateObservation	nbCnx
101	9940001A	2017/05/31	2
102	9940001B	2017/05/31	13
...			

Table HistoCnxEtablissement		
rneEtablissement	dateObservation	nbCnx
9940001A	2017/05/31	15
9940001B	2017/06/30	20
...		

**Document 3 : Extrait du tableau de bord du Lycée Vanille Bourbon
Année scolaire 2015-2016**



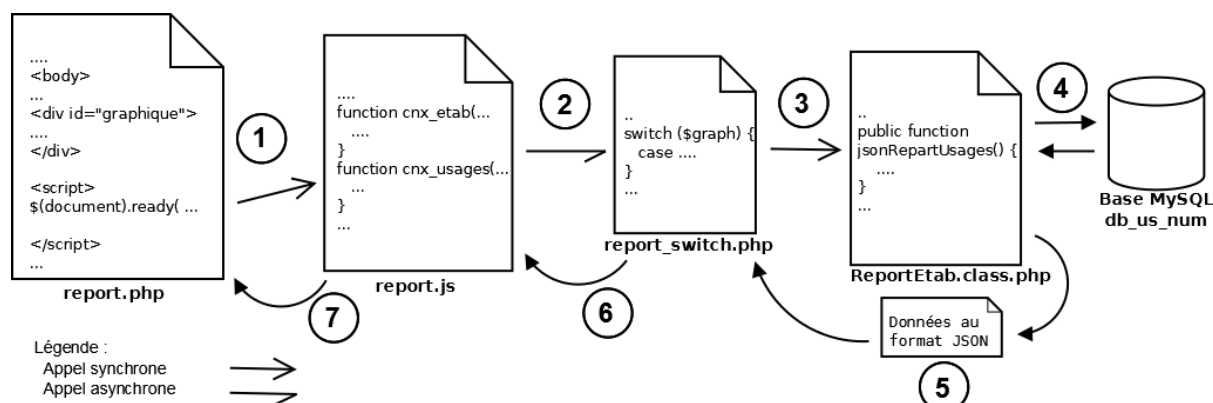
Document 4 : Type de graphique attendu



Ce graphique est basé sur le même jeu de données que dans le document 3.

Il présente la répartition des visites par catégorie d'usages du Lycée Vanille Bourbon sur l'année scolaire 2015-2016 sans tenir compte des usages de type "Vie scolaire" : (*idCategorieUsage* = 108).

Document 5 : Schéma de principe de la génération des graphiques



Explications :

Après avoir sélectionné l'établissement, l'utilisateur est redirigé vers la page d'affichage du tableau de bord nommée **report.php**. Cette page web est adaptative (*RWD responsive web design*), elle s'adapte automatiquement en fonction de la solution d'accès utilisée.

Les graphiques sont positionnés dans des balises `<div>` et un script (fichier **report.js**), chargé « d'alimenter » ces balises, est lancé automatiquement au chargement de la page ①. Ce script lance de manière asynchrone ② le code chargé d'aiguiller le traitement (**report_switch.php**) vers la méthode qui retournera les données attendues ③. Cette méthode interroge la base de données ④ et retourne le résultat au format *JSON* ⑤. Ce résultat est exploité par le script du fichier **report.js** pour créer le graphique ⑥ puis l'afficher dans la page **report.php** ⑦.

Note :

Le format en notation objet *JavaScript* (*JSON*) est un format léger d'échange de données qui permet de structurer un ensemble d'informations au sein d'une seule chaîne de caractères.

Exemple de chaîne retournée par la méthode `ReportEtab::jsonRepartUsages()` :
`[{"usage": "Travail collaboratif", "nb_cnx": "493"}, ... , {"usage": "Orientation", "nb_cnx": "241"}]`

		Page : 9/18

Projet : Usnum V2

Émise le : 26/04/2017

Demande de modification n : 1

Description de la modification demandée

Nom : M. Vans

Fonction : Ingénieur de la DSI

Niveau d'importance : Faible ☐
 Majeur ☒
 Bloquant ☐

Fonctionnalité concernée : Tableau de bord.**Description** : Ajout d'un nouveau graphique au tableau de bord.**Implémentation technique :**

J'ai déjà ajouté les lignes 7 et 19 au fichier **report.php** qui permettent d'afficher le nouveau graphique (**document 4**). A toi de terminer l'implémentation dans les fichiers **report.js**, **report_switch.php** et **ReportEtab.class.php**.

Tu peux t'aider du schéma de principe de la génération des graphiques (**document 5**).

Alain

Extrait du script **report.php** :

```

...
1      <div data-role="collapsible" data-theme="c" data-content-theme="d"
2          data-corners="false" data-collapsed-icon="arrow-d" >
3          <h3>Vue d'ensemble des visites</h3>
4          <div id="graphiques">
5              <div id="graph_cnx_etab" style="max-width: 700px"></div>
6              <div id="graph_cnx_usages" style="max-width: 700px"></div>
7              .....<div id="graph_cnx_usages_horsVS" style="max-width: 700px"></div>
8          </div>
9      </div>
10     ...
11     <script>
12         $(document).ready(function () {
13             rne = '<?php echo $rne; ?>';
14             libType = '<?php echo echo $libType; ?>';
15             dateDebut = '<?php echo $dateDebut; ?>';
16             dateFin = '<?php echo $dateFin; ?>';
17             cnx_etab(rne, libType, dateDebut, dateFin);
18             cnx_usages(rne, dateDebut, dateFin);
19             cnx_usages_horsVS(rne, dateDebut, dateFin);
20         });
21     </script>
22     ...

```

Emplacement des divers graphiques.

Les variables \$rne (code établissement) et \$libType (type établissement), \$dateDebut et \$dateFin sont initialisées au chargement de la page. Elles correspondent aux choix effectués par l'utilisateur.

Appel des fonctions chargées de définir et d'activer l'affichage des graphiques. Ces fonctions sont contenues dans le script report.js

Extrait du script **report.js** :

```

...
1      function cnx_usages(rne, dateDebut, dateFin) {
2
3          var optionsCnxUsages = {};
4          optionsCnxUsages.chart = {
5              renderTo: 'graph_cnx_usages', backgroundColor: '#E6F8E0',
6              borderColor: '#BDBDBD', borderWidth: 3
7          };
8          optionsCnxUsages.title = {
9              text: 'Répartition des visites par catégorie d\'usages', style: {fontWeight:
10                 'bold'}
11          };
12
13          $.getJSON('report_switch.php?action=EtabRecapUsages&rne=' +
14              rne+'&dateDebut='+dateDebut+'&dateFin='+dateFin, function(dataCnxUsages) {
15
16              optionsCnxUsages.data = {total : 0, stats : []};
17              $.each(dataCnxUsages, function(i, e){
18                  optionsCnxUsages.data.total += e.nb_cnx;
19                  optionsCnxUsages.data.stats.push({usage : e.usage, nb_cnx : e.nb_cnx});
20              });
21              chartCnxUsages = new Highcharts.Chart(optionsCnxUsages);
22          });
23      }
24      ...

```

Propriétés du graphique

Nom de la div où placer le graphique

Appel jQuery asynchrone du script qui retourne les données au format JSON.

Fonction jQuery qui boucle sur chaque élément du tableau dataCnxUsages contenant les données JSON désérialisées.

Création et affichage du graphique.

Extrait du script **report_switch.php** :

```
...
require_once 'ReportEtab.class.php';

$action = $_GET['action'];
$rne = $_GET['rne'];
$dateDebut = $_GET['dateDebut'];
$dateFin = $_GET['dateFin'];

$report = new ReportEtab($rne, $dateDebut, $dateFin);
switch ($action) {
    case 'EtabEvolCnx': echo $report->jsonEvolCnx();break;
    case 'EtabRepartUsages': echo $report->jsonRepartUsages();break;
    ...
}
...
```

Récupération des paramètres transmis lors de l'appel jQuery asynchrone.

Aiguillage vers la méthode chargée d'afficher les données attendues au format JSON.

Extrait du script **ReportEtab.class.php** :

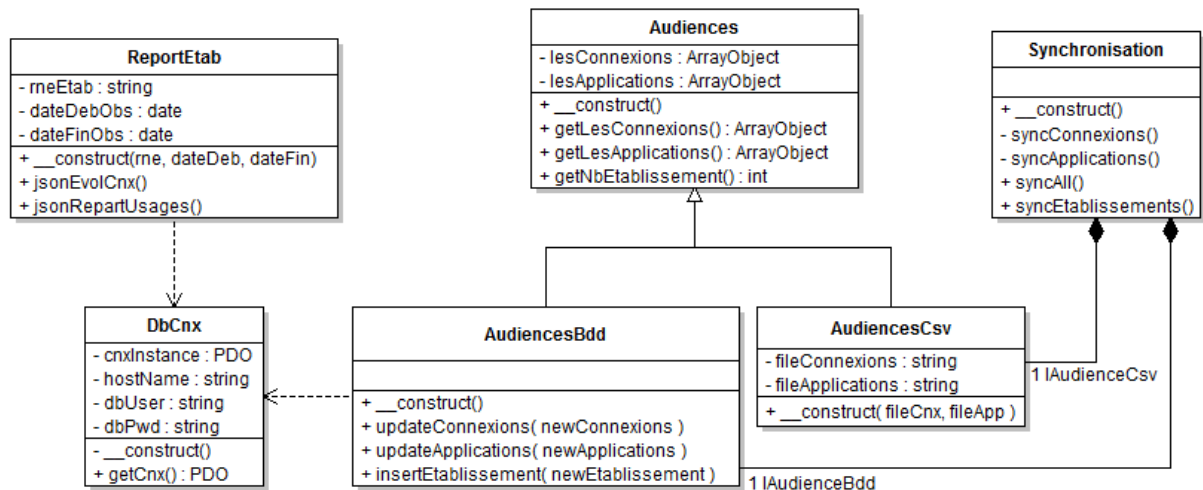
```
1    class ReportEtab {
2        ...
3    public function jsonRepartUsages() {
4        $sql = 'SELECT libelle AS usage, SUM(nbCnx) AS nb_cnx'.
5              ' FROM HistoUsageEtablissement INNER JOIN CategorieUsage '.
6              ' ON id = idCategorieUsage'.
7              ' WHERE rneEtablissement = :rne'.
8              ' AND dateObservation BETWEEN :dateDebut AND :dateFin'.
9              ' GROUP BY libelle';
10       $res = DbCnx::getCnx()->prepare($sql);

11       $res->bindValue(':rne', $this->rne, PDO::PARAM_STR);
12       $res->bindValue(':dateDebut', $this->dateDebObs, PDO::PARAM_STR);
13       $res->bindValue(':dateFin', $this->dateFinObs, PDO::PARAM_STR);
14       $res->execute();
15       $lines = $res->fetchAll(PDO::FETCH_ASSOC);
16       $res->closeCursor();
17       return json_encode($lines);
18   }
19   ...
20 }
```

La connexion à la base est "récupérée" en appelant la méthode de classe getCnx().

Retourne une chaîne de caractères, au format JSON, composée des données renvoyées par la requête.

Document 8 : Diagramme des classes métiers (extraits)



La classe *DbCnx* utilise la technologie objet d'accès aux données *PDO* (*PHP Data Objects*). Elle se charge de la connexion à la base *db_us_num*.

Document 9 : Description partielle des classes métiers utilisées dans le document 8

```

class Audiences {

    private $lesConnexions ;    // Dictionnaire contenant le nombre de connexions
                                // par établissement et par catégorie d'usages

    private $lesApplications ;  // Dictionnaire contenant le nom des applications
                                // activées par établissement et par type d'utilisateur

    public function __construct(){...};

    public function getLesConnexions() {
        return $this->lesConnexions;
    }

    public function getLesApplications() {
        return $this->lesApplications;
    }

    public function getNbEtablissements(){    // retourne le nombre d'établissements
                                                // contenus dans le dictionnaire
                                                // méthode à coder sur votre copie
    }

}
    
```

Les attributs **\$lesConnexions** et **\$lesApplications** sont des instances de la classe *ArrayObject*. Ces dictionnaires ont pour clé le code de l'établissement (*rne*) et pour valeur un tableau dans lequel on trouve le type et le nom de l'établissement suivis de différentes valeurs.

Par exemple voici le contenu du dictionnaire **\$lesConnexions** :

```
object(ArrayObject)#1 (120) {
    ['9940001A']=> array(12) {
        [0]=> string(3) "LYC "
        [1]=> string(46) "Lycée Vanille Bourbon"
        [2]=> int 13007
        [3]=> int 19
        ...
        [11]=> int 151
    }
    ['9940001B']=> array(12) {
        [0]=> string(3) "CLG"
        [1]=> string(15) "Collège Le Busard"
        [2]=> int 12321
        [3]=> int 24
        ...
        [11]=> int 29
    }
    ...
}
```

} nombre de connexions
pour chaque catégorie d'usages

```
class AudiencesCsv extends Audiences {

    private $fileConnexions;    // Nom du fichier CSV contenant les informations
                                // sur les nombres de connexions

    private $fileApplications;  // Nom du fichier CSV contenant les informations
                                // sur les applications activées

    public function __construct( $fileCnx, $fileApp ) {
        $this->fileConnexions = $fileCnx;
        $this->fileApplications = $fileApp;

        // Chargement des fichiers CSV dans les dictionnaires...
    }
}
```

```

class AudiencesBdd extends Audiences {

    public function __construct() {
        // Connexion à la base de données et chargement des données dans les dictionnaires
    }

    public function updateConnexions( $newConnexions ) {
        // Mise à jour des informations à partir du dictionnaire $newConnexions
    }

    public function updateApplications( $newApplications ) {
        // Mise à jour des informations à partir du dictionnaire $newApplications
    }

    public function insertEtablissement( $newEtablissement ) {
        // Effectue l'insertion dans la base de données à partir de la chaîne
        // de caractères $newEtablissement. Ne fait rien si $newEtablissement
        // est une chaîne vide ou une chaîne ne respectant pas le format ci-après.
        // $newEtablissement doit être une chaîne de caractères composée
        // du code rne, du type et du nom de l'établissement séparés par un point-virgule.
        // Par exemple : "990001Z;CLG;Collège Nouveau"
    }
}

```

```

class Synchronisation {

    private $IAudienceCsv; // un objet AudiencesCsv

    private $IAudienceBdd; // un objet AudiencesBdd

    public function __construct() {
        $this->IAudienceCsv = new AudiencesCsv("connexions.csv", "applications.csv");
        $this->IAudienceBdd = new AudiencesBdd();
    }

    private function syncConnexions() {
        $cnx = $this->IAudienceCsv->getLesConnexions();
        $this->IAudienceBdd->updateConnexions($cnx);
    }

    private function syncApplications() {
        $app = $this->IAudienceCsv->getLesApplications();
        $this->IAudienceBdd->updateApplications($app);
    }

    public function syncAll() {
        $this->syncConnexions();
        $this->syncApplications();
    }

    public function syncEtablissements(){
        // teste si les 2 sources de données ont le même nombre d'établissements
        // si ce n'est pas le cas met à jour la base de données en appelant la méthode
        // insertEtablissement de la classe AudiencesBdd pour chaque nouvel établissement
        // Méthode à coder sur votre copie
    }
}

```

Chargement des deux sources de données en mémoire.

Méthodes privées qui réalisent les mises à jour de la base de données à partir des informations présentes dans les fichiers CSV.

Méthode publique qui lance les mises à jour de la base de données.

Cette classe technique permet de gérer une collection de type dictionnaire (ou tableau associatif) : les éléments sont indexés par une clé (index).

ArrayObject implements IteratorAggregate, ArrayAccess, Serializable, Countable {

```

/* Constantes */
const integer STD_PROP_LIST = 1 ;
const integer ARRAY_AS_PROPS = 2 ;

/* Méthodes */
public __construct ([ mixed $input = [] [, int $flags = 0 [, string $iterator_class =
"ArrayIterator" ]]] )

public void ArrayObject::offsetSet ( mixed $cle , mixed $newval )
// Définit $newval comme valeur à la $cle spécifié

public bool ArrayObject::offsetExists ( mixed $index )
// Vérifie si une clé existe

public void ArrayObject::ksort ( void )
// Trie un tableau par clé

public void ArrayObject::asort ( void )
// Trie les éléments par valeur

public int ArrayObject::count ( void )
// Retourne le nombre d'éléments
}

```

Le pseudo-type **mixed** indique que le paramètre peut accepter plusieurs types.

Exemple d'utilisation :

```

<?php
$fruits = array("d" => "citron", "a" => "orange", "b" => "banane", "c" => "pomme");
$fruitArrayObject = new ArrayObject($fruits);
$fruitArrayObject->asort();
echo "Voici les " . $fruitArrayObject->count() . " fruits triés par nom : \n" ;
foreach ($fruitArrayObject as $cle => $valeur) {
    echo "$cle = $valeur\n";
}
?>

```

L'exemple ci-dessus va afficher :

Voici les 4 fruits triés par nom :

b = banane
 d = citron
 a = orange
 c = pomme



Délégation académique
au numérique pour l'éducation



Observation des usages du numérique éducatif

Établissement : Lycée Vanille Bourbon	Référent : JC TEUFAIX
Professeurs concernés : Mme TECHER et M. DUPOND	Date : 06/12/2016
	Classe : 1 ES
Période séquence : du 3 octobre 2016 au 7 novembre 2016	Matière : Mathématiques

Observation/Expérimentation :

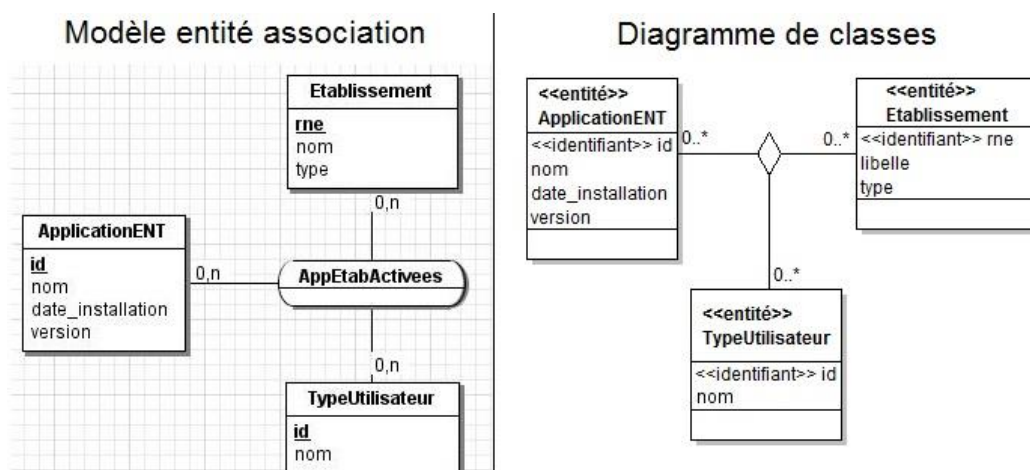
Classe inversée avec capsules vidéo et évaluation formative à l'aide de QCM.

Applications hors ENT		Applications ENT	Matériels	
Nom	Web ¹ (o/n)		Type	Etab ² (o/n)
Socrative	<input checked="" type="checkbox"/>	Moodle	Ordinateur portable	<input checked="" type="checkbox"/>
EDpuzzle	<input checked="" type="checkbox"/>	smartphone	<input type="checkbox"/>
.....	<input type="checkbox"/>	<input type="checkbox"/>
.....	<input type="checkbox"/>	<input type="checkbox"/>
.....	<input type="checkbox"/>	<input type="checkbox"/>

¹ Préciser si l'application a nécessité une connexion à *internet* lors de l'observation

² Indiquer s'il s'agit d'un matériel mis à disposition par l'établissement

Document 12 : Extrait du schéma conceptuel existant fourni dans les deux représentations les plus courantes



Document 13 : Proposition de maquette

☎ 0262 30 35 94 ✉ ce.dane@ac-r.fr
Ac-R Métice WpEtab InfoMétice

Délégation académique
au numérique pour l'éducation
Déconnexion

Observation des usages du numérique éducatif

Etablissement : ①	Professeur référent : ②
Professeurs concernés : ③	Date : ④
Période concernée : ⑥	Classe : ⑤
	Matière : ⑦

Observation / Expérimentation : ⑧

Applications hors ENT		Applications ENT	Matériels	
Nom	web (O/N) ^⑩		Type	Etab (O/N) ^⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬
⑨	⑩	⑪	⑫	⑬

Légende des zones énumérées :

①	Nom de l'établissement, présent dans la base de données
②	Nom du professeur référent, un seul par établissement, présent dans la base de données
③	Le ou les noms des professeurs observés, devront être accessibles depuis la base de données, maximum 2 professeurs
④	Date de saisie de l'observation
⑤	Classe observée, devra être accessible depuis la base de données
⑥	Période concernée par l'utilisation du numérique (date début et date de fin)
⑦	Matière observée, devra être accessible depuis la base de données
⑧	Descriptif de l'observation, prévoir 500 caractères au maximum
⑨	Nom des applications hors <i>ENT</i> utilisées lors de l'observation, devra être accessible depuis la base de données
⑩	Utilisation d'un accès <i>web</i> oui ou non
⑪	Nom des applications <i>ENT</i> utilisées, devra être accessible depuis la base de données
⑫	Matériel utilisé, devra être accessible depuis la base de données
⑬	Matériel fourni par l'établissement oui ou non