

Épreuve spécifique A : Étude de cas informatique

Sujet 0*

Introduction

Présentation du sujet

Le projet proposé s'inspire d'un projet réel présenté en annexe, section B.

La procédure de conception utilisée dans le sujet s'inspire librement de l'approche agile avec une organisation en parties ("sprints"). Chaque partie comprend la définition d'un certain nombre d'objectifs concrets, la réflexion sur les moyens de les atteindre, la préparation d'une procédure de validation permettant de vérifier si les objectifs sont atteints et une discussion critique de l'ensemble du processus.

La première partie est guidée de manière assez précise car elle sert de point de départ à l'ensemble des autres parties. Il est donc recommandé de commencer par cette première partie. Ensuite, les autres parties sont largement indépendantes. Les questions posées sont largement ouvertes.

Vos réponses seront évaluées selon les critères suivants.

- Les choix de conception doivent être conduits en privilégiant la clarté, la robustesse, la cohérence, la portabilité et la simplicité.
- Le processus qui vous a conduit à proposer ces choix doit être expliqué et justifié.
- Les réponses des questions successives doivent décrire un processus de conception cohérent tout au long de votre copie.

Langage de programmation

Il est attendu que vous proposiez une approche orientée objet en utilisant le langage Python. Les concepts utilisables en programmation orientée objet sont : objets, classes, héritage, polymorphisme de sous-typage.

En ce qui concerne la programmation Web, il est attendu que vous utilisez les langages HTML, CSS et JavaScript.

Il est demandé de privilégier la robustesse, la portabilité et la maintenabilité. On pourra discuter à la fin de chaque partie des questions de robustesse logicielle posées par l'utilisation du langage proposé.

*Date de rédaction : 2021-11-15, version : 4cb388b

Partie I Analyse et programmation

Vous préparez une réponse au cahier des charges "Développement d'une page Web dédiée au circuit scientifique bordelais pour la fête de la science 2014".

Une version simplifiée est disponible en annexe, section B. De fait, seules les parties relatives au cycle de développement vous intéressent dans ce cahier des charges.

Le langage de programmation à utiliser dans cette partie est Python.

I.1 Analyse du cahier des charges et conception orientée objet

Classe Créneau

Le cahier des charges est fondé sur la notion de créneau horaire.

Dans le cadre de ce sujet, nous considérons qu'un créneau est défini par un horaire de début, un lieu et une durée.

Pour l'instant, on considère qu'un créneau n'accueille qu'un seul participant, par exemple une classe. Le cas de l'accueil de plusieurs groupes sera traité plus tard.

Chaque créneau est donc soit libre, soit occupé.

Question I.1.1 *Proposez une implémentation de cette classe en précisant soigneusement les types des attributs et les signatures des méthodes. (On ne demande pas de décrire le corps des méthodes.)*

Classe Atelier

Le cahier des charges définit la notion d'atelier.

Dans le cadre de ce sujet, nous considérons qu'un atelier est défini par son identifiant.

Un atelier est attribué à un animateur. Un atelier propose un certain nombre de créneaux. Il garantit que les créneaux proposés sont disjoints. Il permet de modifier le statut de chacun des créneaux qu'il propose.

Pour l'instant, on considère que les ensembles de créneaux affectés aux ateliers sont disjoints deux à deux. Le cas des créneaux en superposition sera traité plus tard.

Question I.1.2 *Proposez une implémentation de cette classe en précisant soigneusement les types des attributs et les signatures des méthodes. (On ne demande pas de décrire le corps des méthodes.)*

Classe Animateur

Le cahier des charges mentionne des animateurs d'ateliers.

Dans le cadre de ce sujet, un animateur est défini par son identifiant.

Il gère un certain nombre d'ateliers. Il peut ajouter un nouvel atelier. La question de la suppression d'un atelier sera traitée plus tard.

L'animateur garantit que les créneaux proposés par tous les ateliers qu'il gère sont disjoints.

Question I.1.3 *Proposez une implémentation de cette classe en précisant soigneusement les types des attributs et les signatures des méthodes. (On ne demande pas de décrire le corps des méthodes.)*

Classe Participant

Le cahier des charges mentionne que les ateliers accueillent des participants. Dans le cadre de ce sujet, un participant est défini par son identifiant. On considère que chaque créneau accueille au plus un participant. Un participant est inscrit à un certain nombre de créneaux d'ateliers qu'il conserve. Il garantit que tous les créneaux où il est inscrit sont disjoints.

Question I.1.4 *Proposez une implémentation de cette classe en précisant soigneusement les types des attributs et les signatures des méthodes. (On ne demande pas de décrire le corps des méthodes.)*

I.2 Programmation détaillée de certaines méthodes

Dans cette section, il est expressément demandé de privilégier la robustesse, la portabilité et la maintenabilité.

Vous veillerez à définir une stratégie de gestion des situations exceptionnelles et à traiter la totalité des situations exceptionnelles selon cette stratégie. L'utilisation judicieuse d'assertions (structure de contrôle assert en Python, etc.) pour garantir la robustesse de votre code sera appréciée.

Affichage de inscriptions à un atelier d'un animateur

Question I.2.1 *Programmez les méthodes nécessaires pour imprimer à l'écran la liste des participants à l'un des ateliers d'un animateur.*

Vous veillerez à gérer le cas où l'animateur n'est en fait pas associé à cet atelier.

Désinscription d'un participant à un atelier

Question I.2.2 *Programmez les méthodes nécessaires pour désinscrire un participant d'un atelier où il est inscrit.*

Vous veillerez à gérer le cas où le participant n'est en fait pas inscrit.

Affichage des créneaux d'un participant

Question I.2.3 *Programmez les méthodes nécessaires pour imprimer la liste des créneaux où un participant est inscrit. Pour chaque créneau, on fournira l'identifiant de l'atelier, l'identifiant de l'animateur, l'horaire de début, le lieu, la durée.*

Vous veillerez à fournir la liste des créneaux par ordre chronologique croissant de début.

Ajout d'un nouveau créneau à un atelier

Question I.2.4 *Programmez les méthodes nécessaires pour ajouter un nouveau créneau à un atelier.*

Vous veillerez à gérer explicitement toutes les situations anormales qui peuvent survenir.

Recherche d'un créneau libre

Question I.2.5 *Programmez les méthodes nécessaires pour imprimer la liste des identifiants de tous les ateliers qui ont un créneau libre à une heure donnée. L'heure est passée en paramètre*

à la fonction. Pour chaque créneau, on indiquera le temps déjà écoulé depuis le début du créneau.

Vous veillerez à détailler la manière dont la liste de tous les ateliers est produite.

I.3 Scénarios de référence et tests fonctionnels

Il s'agit ici de présenter quelques scénarios, c'est-à-dire des séquences d'interaction typiques qui pourront servir de base pour les tests fonctionnels.

Dans chaque scénario, précisez dans un commentaire l'état des objets impliqués, au début et à la fin du scénario. Pour chaque interaction du scénario, précisez l'objet qui a déclenché cette interaction et celui qui est sollicité.

Scénarios de création

Question I.3.1 Proposez un scénario qui montre la création pour un animateur d'un atelier avec un unique créneau.

Question I.3.2 Proposez un scénario qui montre la création d'un atelier avec deux créneaux non disjoints.

Scénarios d'inscription

Question I.3.3 Proposez un scénario qui montre l'inscription d'un participant à un atelier.

Question I.3.4 Proposez un scénario qui montre l'inscription d'un participant à un atelier à un créneau donné alors que ce participant est déjà inscrit à un autre atelier sur un créneau non disjoint

Scénarios d'annulation

Question I.3.5 Proposez un scénario qui inclut la désinscription d'un participant à un atelier.

Partie II Interface Web

Les langages de programmation à utiliser dans cette partie sont HTML, CSS et JavaScript.

II.1 Formulaire d'inscription à un atelier (HTML et formulaire)

La page nouvelle_inscription.html est présentée sur la copie d'écran en figure 1

Elle permet à un participant de s'inscrire à un atelier. Chaque atelier est repéré par un identifiant. Les participants ont accès par ailleurs à la liste des identifiants des ateliers.

Cette page Web contient un formulaire permettant au participant de renseigner l'identifiant de l'atelier auquel il souhaite s'inscrire. Ce formulaire soumet une requête HTTP POST vers /inscrire_atelier. Il contient un seul champ nommé code_atelier.

Question II.1.1 Proposez une programmation en HTML pur de l'ensemble de cette page, sans JavaScript. On ne traitera pas les éventuels retours d'erreur de la requête.

FIGURE 1 – La page nouvelle_inscription.html.

II.2 Gestion des inscriptions (JavaScript et CSS)

La page inscriptions.html est présentée dans la table 2.

Elle permet à un participant de se désinscrire d'un atelier où il s'est inscrit.

La désinscription à un atelier donné suit la procédure suivante.

- Si l'atelier est planifié dans plus de 2 jours, il suffit de cliquer sur le bouton associé intitulé Se désinscrire.
- Si l'atelier est planifié dans moins de 2 jours, il n'est pas possible de se désinscrire directement sur la page Web. Le participant doit téléphoner au service du CNRS pour l'informer qu'il annule son inscription.

Cette procédure est implémentée par un script JavaScript porté par la page HTML. Par défaut, la page associe un bouton Se désinscrire à chaque atelier.

Si la date de l'atelier est trop proche, le bouton associé à cet atelier doit être remplacé par une zone de texte : Appelez le numéro 01 400 400 pour vous désinscrire.

Ceci est implémenté en modifiant dynamiquement le DOM de la page à son chargement. On remplace dans le DOM les boutons des ateliers concernés par la zone de texte.

Le code HTML de la page est fourni en Annexe, section A.

Question II.2.1 Proposez un script JavaScript qui réalise cette transformation.

II.3 Formulaire d'inscription (JavaScript)

Lorsque la désinscription est demandée deux jours ou plus avant le déroulement de l'atelier, elle est automatiquement gérée par l'application, du côté du serveur.

Le code unique de l'inscription est défini dans l'attribut id_inscription dans la page HTML. Un exemple de code est 456534RZZ.

Un clic sur le bouton Se désinscrire envoie une requête HTTP GET vers l'URL qui commence par

`http://www.serveur.fr/desinscription/`

et qui se termine par le code de l'inscription. Par exemple, envoyer une requête GET à l'adresse

`http://www.serveur.fr/desinscription/5456534RZZ`

permet de se désinscrire de l'atelier 5456534RZZ.

Vos inscriptions aux ateliers

Titre Liquides durs et solide mous
Date 3/10/2021
Horaire 9 h
Laboratoire Centre de Recherche Paul Pascal, CNRS
Lieu 115 Avenue du Dr Schweitzer, 33600 Pessac
Capacité 12 personnes
Durée 30 mn
Confirmation oui

[Se désinscrire](#)

Titre Cybersécurité
Date 3/10/2021
Horaire 10 h
Laboratoire LaBRI
Lieu Domaine universitaire, 351, cours de la Libération, 33405 Talence
Capacité 12 personnes
Durée 30 mn
Confirmation non

[Se désinscrire](#)

FIGURE 2 – La page inscriptions.html. Voir le code source en annexe A.

Question II.3.1 Proposez un nouveau script JavaScript pour la page "inscriptions.html" pour permettre l'envoi d'une telle requête GET lorsqu'on clique sur un bouton *Se désinscrire*.

Si la réponse à la requête est positive (code de retour : 200), le script devra supprimer dans le DOM la partie correspondant à l'inscription ainsi annulée. Si la réponse à la requête est négative (code de retour différent de 200), le script devra lancer une alerte précisant que la suppression n'a pas été réalisée.

Indication. On rappelle que la fonction JavaScript `alert(message)` permet de lancer une alerte à l'utilisateur. L'utilisateur n'a plus accès à la page initiale tant que l'alerte n'a pas été acquittée.

Partie III Raffinement et optimisation

Dans cette partie, il s'agit de reprendre la conception de votre application pour lui permettre de traiter un plus grand nombre de requêtes, à la fois en nombre d'ateliers, en nombre de créneaux, en nombre d'animateurs et en nombre de participants.

Le langage de programmation à utiliser dans cette partie est Python.

On s'intéresse au coût en temps et en espace. Pour le coût en temps, on pourra par exemple prendre comme unité le coût d'une comparaison de deux valeurs manipulées par l'application, par exemple des identifiants ou des horaires.

Pour vous donner un ordre de grandeur de dimensionnement, il y a quelques centaines d'établissements d'enseignement supérieur en France, qui proposent potentiellement chacun plusieurs dizaines d'ateliers une dizaine de fois par jour sur une semaine. Chaque atelier peut attirer quelques dizaines de personnes.

III.1 gestion des créneaux

Plusieurs solutions sont possibles pour gérer les créneaux et garantir leurs propriétés.

- Les créneaux associés à un animateur donné doivent être disjoints
- Les créneaux associés à un atelier donné doivent être disjoints

Question III.1.1 Proposez une étude du coût en temps et en espace de votre implémentation des tables de créneaux associées à un animateur ou un atelier.

- Coût en temps de l'ajout d'un créneau pour un atelier ou pour un animateur.
- Coût en temps de l'effacement d'un créneau pour un atelier ou pour un animateur.
- Coût en temps pour qu'un participant trouve dans un atelier donné un créneau libre dans ses plages de disponibilité.
- Coût en mémoire du stockage de l'ensemble des tables de créneaux pour tous les ateliers et tous les animateurs.

Ces coûts en temps et en espace sont-ils satisfaisants lorsqu'il y a un grand nombre d'ateliers et d'animateurs? Par exemple, on peut supposer qu'il y a beaucoup plus de consultations que d'insertions et beaucoup plus d'insertions que d'effacements.

III.2 Gestion de l'inscription des groupes

Dans une première version de l'application adaptée à la participation de classes, on peut considérer que les participants aux ateliers sont des classes, inscrites par leurs professeurs. Il n'y a donc qu'un seul participant pour chaque créneau, le professeur qui fait l'inscription de sa classe.

On s'intéresse maintenant au cas des petits groupes, typiquement des familles. Chaque créneau offre un certain nombre de places. Un groupe qui s'inscrit précise le nombre de personnes qui le composent en plus de ses disponibilités.

De plus, le groupe peut préciser un ordre de priorité sur ses créneaux disponibles. D'un autre côté, l'application cherche à compléter les créneaux déjà occupés et à les regrouper sur des plages horaires contiguës pour un même animateur.

Question III.2.1 *Proposez un raffinement de votre implémentation qui prenne en compte cet aspect. Précisez soigneusement vos critères d'optimisation.*

III.3 Gestion du temps de transport

Les ateliers sont proposés dans différents lieux. On dispose d'une table qui fournit le temps de transport à prendre en compte entre deux lieux donnés à une heure donnée. Cette table n'est pas forcément symétrique : aux heures de pointe, les embouteillages ne sont souvent que dans une seule direction !

Il faut donc garantir que l'allocation des créneaux respecte ces contraintes de temps, à la fois pour les participants et les animateurs. Deux créneaux successifs doivent être espacés d'assez de temps pour se transporter d'un lieu à l'autre.

Question III.3.1 *Proposez un raffinement de votre implémentation qui prenne en compte cet aspect.*

III.4 Modification à la volée des inscriptions

Si un animateur annule un créneau, il faut essayer de trouver des créneaux intéressants à proposer aux participants de ce créneau, sur le même lieu et à peu près aux mêmes horaires.

On dispose d'une table d'affinité entre ateliers. L'affinité est une valeur entre 0 et 1. Une affinité de 1 signifie que les deux ateliers sont proches thématiquement. Les participants à l'un seront sans doute intéressés par l'autre. Une affinité de 0 signifie que les thèmes des deux ateliers n'ont rien à voir.

Notez que les propositions doivent être globalement compatibles. Si chacun des participants accepte la proposition d'inscription instantanément, aucune de ces nouvelles inscriptions ne doit être rejetée par l'application.

Question III.4.1 *Proposez une fonction qui fournit des propositions intéressantes pour tous les participants d'un créneau annulé. L'identifiant du créneau est passé en paramètre.*

Partie IV Gestion des données

Le langage de programmation à utiliser dans cette partie est Python.

Selon les ordres de grandeur ci-dessus, les données de l'application sont potentiellement partagées par des milliers d'utilisateurs. Il est donc indispensable de gérer les conflits d'accès à ces données.

IV.1 Gestion des identifiants et tolérance aux erreurs

Chaque atelier, chaque créneau, chaque participant, chaque animateur est associé à un identifiant unique. Le choix de la forme de ces identifiants est important puisque l'application va manipuler ces identifiants de manière répétée.

Question IV.1.1 *Proposez un mécanisme de production et d'allocation de ces identifiants. Il doit garantir l'unicité des identifiants.*

Précisez le coût de stockage de vos identifiants. Précisez le coût de la comparaison de deux tels identifiants lors d'une recherche dans une table en fonction du nombre maximal d'identifiants à gérer dans l'application.

IV.2 Gestion partagée des données

L'application peut être utilisée de manière simultanée par de nombreux utilisateurs qui s'inscrivent ou se désinscrivent sur des créneaux d'ateliers. Il serait pénalisant de traiter ces interactions de manière bloquante, une inscription ne commençant que lorsque la précédente est terminée.

Il faut donc prévoir un mécanisme de gestion partagée des données qui permette à plusieurs utilisateurs d'y accéder en même temps tout en garantissant le maintien de la cohérence globale. L'objectif est d'optimiser le délai d'attente des utilisateurs lors d'une requête.

Question IV.2.1 *Proposez un tel mécanisme de gestion partagée des données. Évaluez la réduction du délai d'attente pour les utilisateurs selon les requêtes.*

Notez qu'il n'est pas demandé de programmer entièrement ce mécanisme mais seulement de le décrire suffisamment précisément pour que sa correction puisse être établie et que son impact puisse être évalué.

IV.3 Notion de réservation

Un utilisateur qui accède à l'interface définit ses possibilités horaires et reçoit en retour une proposition d'inscription. Il a ensuite quelques minutes pour l'accepter ou pas. Il faut donc que cette proposition reste valide durant ce délai, quelles que soient les actions des autres utilisateurs.

De même, si un créneau est annulé, l'application renvoie aux participants une proposition alternative, à valider dans les 24 heures. Il faut donc que cette proposition reste valide durant ce délai, quelles que soient les actions des autres utilisateurs.

Il est donc nécessaire d'introduire le concept de réservation avec une date limite de validité. Tant que la date n'est pas atteinte, l'application garantit la réservation. L'utilisateur peut la valider. Une fois la date atteinte, la réservation peut être effacée par l'application sans avertissement.

Question IV.3.1 *Proposez un tel mécanisme. Évaluez le coût additionnel global induit par son introduction.*

Notez qu'il n'est pas demandé de programmer entièrement ce mécanisme mais seulement de le décrire suffisamment précisément pour que sa correction puisse être établie et que son impact puisse être évalué.

Partie V Base de données SQL

L'approche présentée en partie I était fondée sur la programmation orientée objet. On se propose d'explorer une approche fondée sur des bases de données.

Le langage à utiliser dans cette partie est SQL.

Indication. *On pourra librement utiliser des tables intermédiaires pour favoriser la clarté et la robustesse de la conception. On commenterà alors l'efficacité du programme obtenu et on expliquera comment on pourrait éventuellement le transformer pour le rendre plus efficace.*

Dans cette nouvelle approche, l'état courant de l'organisation est définie par les tables suivantes.

Atelier. Un atelier est défini par son identifiant unique (clé primaire) et l'animateur auquel il est attribué.

Créneau. Un créneau est défini par son identifiant unique (clé primaire), un horaire de début, un lieu et une durée et l'unique atelier auquel il est rattaché.

Participant. Un participant est défini par son identifiant unique (clé primaire).

Animateur. Un animateur est défini par son identifiant unique (clé primaire).

Inscription. Cette table contient l'ensemble de toutes les affectations enregistrées. Chaque affectation est définie par identifiant unique (clé primaire), son animateur, son participant et son créneau. Cette table doit garantir qu'un créneau donné n'est affecté qu'une fois au plus.

V.1 Recherche dans les tables

Question V.1.1 Proposez une série de une ou plusieurs requêtes SQL pour effectuer les tâches suivantes.

1. Lister les créneaux proposés par un animateur donné, quel que soient leur statut
2. Lister les créneaux réservés d'un animateur donné, avec pour chacun l'horaire de début, la durée, l'atelier et le participant
3. Renvoyer la durée complète des créneaux réservés pour l'ensemble des animateurs et des participants
4. Vérifier que chaque créneau est affecté au plus une fois.

V.2 Vérification de la cohérence des affectations

Question V.2.1 Proposez une série de une ou plusieurs requêtes SQL pour vérifier que tous les créneaux rattachés à un animateur donné sont deux à deux disjoints.

Question V.2.2 Précisez le coût de votre requête en temps et en espace en fonction des capacités spécifiques de la plate-forme de base de données utilisée.

Question V.2.3 Proposez un jeu de test permettant de vérifier la validité de votre proposition.

V.3 Détermination d'une nouvelle affectation

On permet maintenant aux participants de spécifier des plages de disponibilité. Quand un participant demande à s'inscrire à un atelier, le créneau choisi doit être inclus entièrement dans les plages de disponibilité de ce participant.

Question V.3.1 Ajustez la modélisation pour prendre en compte cette extension.

Question V.3.2 Proposez une série de une ou plusieurs requêtes SQL qui détermine l'ensemble des créneaux possibles pour une demande de réservation d'un participant donné avec des plages de disponibilité données à un atelier donné.

Question V.3.3 Proposez un critère de qualité qui permette de choisir le créneau le plus adéquat pour le participant et pour l'animateur.

Question V.3.4 Proposez un jeu de test permettant de vérifier la validité de votre proposition.

Annexes

A Désinscription à un atelier

```
<!DOCTYPE html>
<html>

<head>
  <link type="text/css" rel="stylesheet" href="inscription.css">
  <title>Vos inscriptions</title>
</head>

<body>
  <h1 id="inscription-planning">Vos inscriptions aux ateliers </h1>
  <div id="inscription-list">
    <div class="inscription" id="inscription-54565342RZZ">
      <table>
        <tr>
          <td>Titre</td>
          <td>Liquides durs et solide mous</td>
        </tr>
        <tr>
          <td>Date</td>
          <td>3/10/2021</td>
        </tr>
        <tr>
          <td>Horaire</td>
          <td>9 h</td>
        </tr>
        <tr>
          <td>Laboratoire</td>
          <td>Centre de Recherche Paul Pascal, CNRS</td>
        </tr>
        <tr>
          <td>Lieu</td>
          <td>115 Avenue du Dr Schweitzer, 33600 Pessac</td>
        </tr>
        <tr>
          <td>Capacité</td>
          <td>12 personnes</td>
        </tr>
        <tr>
          <td>Durée</td>
          <td>30 mn</td>
        </tr>
        <tr>
          <td>Confirmation</td>
          <td>oui</td>
        </tr>
      </table>
      <button>Se désinscrire</button>
    </div>
  </div>
  <hr />
```

```

<div class="inscription" id="inscription="45265342RZZ">
  <table>
    <tr>
      <td>Titre</td>
      <td>Cybersécurité</td>
    </tr>
    <tr>
      <td>Date</td>
      <td>3/10/2021</td>
    </tr>
    <tr>
      <td>Horaire</td>
      <td>10 h</td>
    </tr>
    <tr>
      <td>Laboratoire</td>
      <td>LaBRI</td>
    </tr>
    <tr>
      <td>Lieu</td>
      <td>Domaine universitaire, 351, cours de la Libération, 33405 Talence</td>
    </tr>
    <tr>
      <td>Capacité</td>
      <td>12 personnes</td>
    </tr>
    <tr>
      <td>Durée</td>
      <td>30 mn</td>
    </tr>
    <tr>
      <td>Confirmation</td>
      <td>non</td>
    </tr>
  </table>
  <button>Se désinscrire</button>
</div>
</div>
</body>
</html>

```

B Cahier des charges de l'application étudiée

(Voir page suivante.)

Développement d'une page Web dédiée au circuit scientifique bordelais pour la fête de la science 2014

Document technique (extraits)

Article 1. Objet du marché

1.1 Objet du marché

Le présent marché a pour objet le développement d'un site web dédié au Circuit Scientifique Bordelais, pour la Fête de la Science au CNRS, qui se déroule chaque année sur une semaine pendant la première quinzaine d'octobre.

Article 3. Cahier des charges spécifique à la prestation du marché

3.1 Contexte de la prestation

La Délégation Aquitaine du CNRS recherche un prestataire de service pour le développement de site web dans le but de faciliter l'organisation de l'évènement « Circuit Scientifique Bordelais ».

Ce circuit est l'occasion pour les laboratoires aquitains d'ouvrir leurs portes pendant une semaine au public.

Une vingtaine de laboratoires proposent des ateliers pour découvrir le monde de la recherche de manière ludique. La dernière édition a rassemblé plus de 1500 visiteurs pour une cinquantaine d'ateliers proposés. Les participants sont principalement des élèves du secondaire avec leurs enseignants.

Ce circuit est organisé depuis 16 ans mais face à des contraintes logistiques, le CNRS souhaite créer un site web pour faciliter la mise en place de cet évènement.

L'an passé, le CNRS et le rectorat ont travaillé ensemble dans la planification de ce circuit. Le CNRS a mobilisé les laboratoires, le rectorat a mobilisé les enseignants et s'est occupé de faire correspondre les demandes des enseignants avec les disponibilités des laboratoires.

Les objectifs de ce nouveau site web sont :

- D'attirer plus de public (scolaire et non scolaire) sur le Circuit grâce à une interface facile d'accès
- De faciliter la gestion de inscriptions et à terme la gestion du circuit

3.2 Publics concernés

Les principaux publics concernés sont :

- Les enseignants du Secondaire,
- Les étudiants du Secondaire,
- Le grand public,
- La communauté scientifique.

3.4 Crédit graphique

Les éléments de charte graphique seront communiqués au prestataire lors de la première réunion de cadrage.

3.5 Caractéristiques et objectifs du site

Le CNRS souhaite qu'un site web soit créé et qu'il serve de plate-forme à la fois pour les laboratoires souhaitant ouvrir leurs portes mais aussi pour les inscriptions des enseignants. Ce site web doit notamment comporter les fournitures suivantes.

- La possibilité pour les laboratoires (animateurs) de créer en ligne une fiche synthétique de leur atelier (appelé Atelier) qui ne sera affichée qu'après acceptation par le service communication du CNRS
- La possibilité de rendre cette fiche accessible aux enseignants en proposant plusieurs classements : par laboratoire, par secteur géographique ou encore par thématique.
- La possibilité pour les animateurs d'ateliers de créer et de supprimer des créneaux d'animation (appelé CreneauAtelier)
- La possibilité pour les enseignants d'avoir accès en temps réel à ces disponibilités et pour s'inscrire à un créneau (appelé InscriptionCreneauAtelier)
- La création d'une carte "géographique" localisant les ateliers, voir un calcul du temps qui sépare géographiquement deux ateliers (ou deux "zones")
- Que cette carte géographique soit accessible à l'enseignant en cours d'inscription pour qu'il visualise les choix d'ateliers au fur et à mesure
- La possibilité pour un laboratoire d'accepter ou de refuser une inscription à un atelier.
- La possibilité, pour un enseignant, d'avoir accès à un récapitulatif de ses inscriptions (en cours d'acceptation, acceptées ou refusées) : les ateliers, les contacts, les horaires, les trajets, etc.
- La possibilité, pour les animateurs, d'avoir un récapitulatif des créneaux occupés, des contacts, etc.

Article 4. Exécution des prestations

4.4 Suivi de l'exécution

Les prestations et les opérations de vérification qualitative et quantitative sont effectuées par le CNRS selon les modalités suivantes définies par des livraisons clés de l'opération.

Première livraison

Les laboratoires doivent potentiellement pouvoir déposer la présentation des ateliers et la proposition d'horaire d'accueil du public. Ils doivent avoir la possibilité de changer ces informations à tout moment.

Deuxième livraison

Les enseignants doivent être en mesure de voir le site, d'avoir accès aux différents ateliers et aux différents plannings. Ils doivent pouvoir s'y inscrire. Les laboratoires doivent pouvoir être avertis de cette demande d'inscription et doivent pouvoir valider la demande.

L'enseignant doit être averti de la validation ou du refus.

Troisième livraison

La création de carte géographique regroupant les différents ateliers par zones doit être terminée.