****

Mouhsine El khattab, Marie-Sophie Morin

Dyaa Abou Arida & Christian Sfeir

Création d’une plateforme de gestion du club de secourisme de BdeB

Travail présenté à

M. Raouf Babari

Collège de Bois-de-Boulogne

420-204-RE PROJET D'INTÉGRATION EN SCIENCES INFORMATIQUES

gr. 00002

Le 5 mai 2019

Table des matières

[Création d’une plateforme de gestion du club de secourisme de BdeB 3](#_Toc7971971)

[1. Description du projet 3](#_Toc7971972)

[1.1 Membres de l’équipe 3](#_Toc7971973)

[1.2 Idée 3](#_Toc7971974)

[1.3 Utilité 3](#_Toc7971975)

[1.4 Innovation 4](#_Toc7971976)

[1.5 Lien avec les autres matières 4](#_Toc7971977)

[1.6 Cas d’utilisation 5](#_Toc7971978)

[2. Technologies 8](#_Toc7972159)

[2.1 Technologies à utiliser 10](#_Toc7972160)

[2.2 Défis et difficultés 10](#_Toc7972161)

[2.2.1 Serveur 10](#_Toc7972162)

[2.2.2 Base de données 10](#_Toc7972164)

[2.2.3 Html 11](#_Toc7972166)

[2.2.4 Logique 11](#_Toc7972168)

[2.2.5 Temps 12](#_Toc7972170)

[2.2.6 Travail d’équipe 12](#_Toc7972172)

[3. Perspectives 12](#_Toc7972173)

[4. Plan 13](#_Toc7972174)

[4.1 Diagramme de Gantt 13](#_Toc7972175)

[14](#_Toc7972177)

[4.2 Analyse 15](#_Toc7972178)

[4.3 UML 16](#_Toc7972179)

[5. Matériel de référence 29](#_Toc7972180)

[5.1 Toturiels 29](#_Toc7972181)

[5.2 Tutoriels vidéo 30](#_Toc7972183)

Création d’une plateforme de gestion du club de secourisme de BdeB

1. Description du projet
   1. Membres de l’équipe

|  |  |
| --- | --- |
| Noms, prénoms | DA |
| Marie-Sophie Morin | 1634388 |
| Dyaa Abou Arida | 1655416 |
| Christian Sfeir | 1629045 |
| Mouhsine El Khattab | 1747174 |

* 1. Idée

Nous avons créé une plateforme interactive sous forme de site internet qui soutiendra l’administration du club de secourisme bénévole du cégep de Bois de Boulogne. Le site sera accessible aux coordonnateurs et protéger par un mot de passe. Les coordonnateurs pourront utiliser le site pour générer l’horaire des secouristes bénévoles en début de session, comptabiliser les heures effectuées en fin de session et ajuster l’inventaire.

* 1. Utilité

Notre projet permet d’alléger la charge de travail des coordonnateurs en plus d’améliorer l’organisation générale du club en centralisant l’ajustement de l’inventaire et la comptabilisation des heures ainsi qu’en délestant les coordonnateurs de la lourde tâche d’analyser les disponibilités des secouristes dans le but de générer un horaire optimal respectant de multiples restrictions et conditions.

* 1. Innovation

Notre site répond au grand besoin de réforme organisationnelle que l’équipe de gestion du club nécessite. La plateforme facilite de multiples aspects du travail des coordonnateurs d’une manière innovatrice.

* 1. Lien avec les autres matières

Notre projet combine la programmation informatique et l’implication bénévole au sein du cégep.  Nous avons utilisé les bases en programmation java que nous avons acquises au sein des trois cours d’informatiques du programme science informatique et mathématique pour nous aider dans notre apprentissage de nouveaux langages et pour créer la logique derrière les fonctionnalités du site. Bien que le programme de secourisme ne soit pas un cours, il est une partie intégrante du programme de bénévolat disponible au cégep de Bois-de-Boulogne et encadré par celui-ci. Il permet d’impliquer les étudiants dans une activité qui garde à jour leurs connaissances en secourisme pour certains et en incite d’autres à développer les leurs en participant à la formation de premier soins général fournie par le cégep dans le but de joindre l’équipe par la suite. Les étudiants du club effectuent un bénévolat utile au bon fonctionnement du cégep en assurant une sécurité médicale constante pour leurs camarades de cours. Les coordonnateurs développent leurs habiletés d’organisation et leur talent en leadership d’une manière qui les aidera plus tard sur le marché du travail. Les personnes totalisant 60 heures de secourisme bénévole se voient gratifiés par l’obtention du Bulletin+ à la fin de la session.

* 1. Cas d’utilisation

Avec les paramètres suivants

|  |  |
| --- | --- |
| Variable | Valeur |
| nbHeureMaximallParSemaine | 5 |
| nbHeuresMinShift | 1.5 |
| heureDebutHoraire | 8 |
| heureFinHoraire | 16.5 |
| maximumSecouristeIgnace | 3 |
| maximumSecouristeStPaul | 2 |

Horaire à 2 secouristes avec des shifts trop courts pour être admissibles et secouristes prioritaires

|  |  |
| --- | --- |
| Disponibilités entrées | Horaire produit |
| {{8.0, 8.0, 0.0, 0.0, 8.0}, //s0  {9.0, 10.0, 0.0, 0.0, 9.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {8.5, 0.0, 0.0, 8.0, 10.0}, //s1  {9.5, 0.0, 0.0, 12.0, 12.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0}}; | IGNACE, 2 shifts assignés   |  |  |  |  | | --- | --- | --- | --- | | Secouriste | Jour | Debut shift | Fin shift | | 0 | 1 | 8.0 | 10.0 | | 1 | 3 | 8.0 | 12.0 |   STPAUL, aucun shift assigné  Nb d’heures par secouriste   |  |  | | --- | --- | | Secouriste | Nb D’heures | | 0 | 2 | | 1 | 4 | |

Horaire à 10 secouristes sans erreur et sans cas prioritaire

|  |  |
| --- | --- |
| Disponibilités entrées | Horaire produit |
| {{8.0, 8.0, 0.0, 0.0, 8.0}, //s0  {9.5, 10.0, 0.0, 0.0, 9.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {8.5, 9.0, 0.0, 9.0, 10.0}, //s1  {13.0, 13.0, 0.0, 11.5, 11.0},  {14.5, 15.0, 0.0, 0.0, 0.0},  {16.5, 16.0, 0.0, 0.0, 0.0},  {10.0, 8.0, 10.0, 13.0, 9.0}, //s2  {13.0, 10.0, 11.0, 15.0, 10.0},  {8.0, 0.0, 13.0, 0.0, 0.0},  {10.0, 0.0, 16.0, 0.0, 0.0},  {8.0, 10.0, 11.0, 9.0, 11.0}, //s3  {10.0, 14.0, 13.0, 11.0, 13.0},  {15.0, 0.0, 15.0, 0.0, 14.0},  {16.5, 0.0, 16.5, 0.0, 16.5},  {8.0, 13.0, 8.0, 10.0, 8.0}, //s4  {10.0, 15.0, 11.0, 14.0, 10.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {8.0, 8.0, 0.0, 0.0, 0.0}, //s5  {10.0, 10.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {8.5, 9.0, 0.0, 9.0, 10.0}, //s6  {13.0, 13.0, 0.0, 11.5, 11.0},  {14.5, 15.0, 0.0, 0.0, 0.0},  {16.0, 16.0, 0.0, 0.0, 0.0},  {10.0, 8.0, 10.0, 13.0, 9.0}, //s7  {13.0, 10.0, 11.0, 15.0, 10.0},  {0.0, 0.0, 13.0, 0.0, 0.0},  {0.0, 0.0, 16.0, 0.0, 0.0},  {8.0, 10.0, 11.0, 9.0, 11.0}, //s8  {10.0, 14.0, 13.0, 11.0, 13.0},  {15.0, 0.0, 15.0, 0.0, 14.0},  {16.5, 0.0, 16.5, 0.0, 16.5},  {8.0, 13.0, 8.0, 10.0, 8.0}, //s9  {10.0, 15.0, 11.0, 14.0, 10.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0}}; | IGNACE, 15 shifts assignés   |  |  |  |  | | --- | --- | --- | --- | | # Secouriste | Jour | Début Shift | Fin Shift | | 0 | 0 | 8.0 | 9.5 | | 0 | 1 | 8.0 | 10.0 | | 2 | 0 | 8.0 | 10.0 | | 7 | 0 | 10.0 | 13.0 | | 1 | 1 | 9.0 | 13.0 | | 4 | 1 | 13.0 | 15.0 | | 9 | 2 | 8.0 | 11.0 | | 3 | 2 | 11.0 | 13.0 | | 3 | 3 | 9.0 | 11.0 | | 2 | 3 | 13.0 | 15.0 | | 4 | 4 | 8.0 | 10.0 | | 8 | 4 | 11.0 | 13.0 | | 8 | 4 | 14.0 | 13.0 |   STPAUL, 6 shifts assignés   |  |  |  |  | | --- | --- | --- | --- | | # Secouriste | Jour | Début Shift | Fin Shift | | 5 | 0 | 8.0 | 10.0 | | 6 | 0 | 14.5 | 16.0 | | 5 | 1 | 8.0 | 10.0 | | 8 | 2 | 11.0 | 13.0 | | 8 | 2 | 15.0 | 16.5 | | 6 | 3 | 9.0 | 11.5 |   Nb d’heures par secouriste   |  |  |  |  | | --- | --- | --- | --- | | Sec. | Nb H | Sec. | Nb H | | 0 | 3.5 | 5 | 4.0 | | 1 | 4.0 | 6 | 4.0 | | 2 | 4.0 | 7 | 3.0 | | 3 | 4.0 | 8 | 4.5 | | 4 | 4.0 | 9 | 3.0 | |

Horaire à 2 secouristes avec données erronés

|  |  |
| --- | --- |
| Disponibilités entrées | Horaire produit |
| {{8.0, 10.0, 0.0, 0.0, 8.0}, //s0  {9.0, 10.0, 0.0, 11.0, 9.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {10.5, 0.0, 0.0, 8.0, 10.0}, //s1  {9.5, 0.0, 0.0, 10.0, 12.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0}}; | IGNACE, 2 shifts assignés   |  |  |  |  | | --- | --- | --- | --- | | Secouriste | Jour | Debut shift | Fin shift | | 1 | 3 | 8.0 | 10.0 | | 1 | 4 | 10.0 | 12.0 |   STPAUL, aucun shift assigné  Nb d’heures par secouriste   |  |  | | --- | --- | | Secouriste | Nb D’heures | | 0 | 0 | | 1 | 4 | |

Avec les paramètres suivants

|  |  |
| --- | --- |
| Variable | Valeur |
| nbHeureMaximallParSemaine | 8 |
| nbHeuresMinShift | 1.5 |
| heureDebutHoraire | 8 |
| heureFinHoraire | 16.5 |
| maximumSecouristeIgnace | 3 |
| maximumSecouristeStPaul | 2 |

Horaire à 1 secouriste sans données erronées

|  |  |
| --- | --- |
| Disponibilités entrées | Horaire produit |
| {{8.0, 10.0, 0.0, 8.0, 8.0}, //s0  {10.0, 12.0, 0.0, 11.0, 9.0},  {0.0, 14.0, 0.0, 0.0, 13.0},  {0.0, 16.0, 0.0, 0.0, 15.0}}; | IGNACE, 2 shifts assignés   |  |  |  |  | | --- | --- | --- | --- | | Secouriste | Jour | Debut shift | Fin shift | | 0 | 0 | 8.0 | 10.0 | | 0 | 1 | 10.0 | 12.0 | | 0 | 1 | 14.0 | 16.0 |   STPAUL, aucun shift assigné  Nb d’heures par secouriste   |  |  | | --- | --- | | Secouriste | Nb D’heures | | 0 | 6.0 | |

Horaire à 5 secouristes (nombre impair sans cas prioritaire

|  |  |
| --- | --- |
| Disponibilités entrées | Horaire produit |
| {{8.0, 10.0, 0.0, 8.0, 8.0}, //s0  {10.0, 12.0, 0.0, 11.0, 9.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {10.5, 0.0, 0.0, 8.0, 10.0}, //s1  {13.5, 0.0, 0.0, 10.0, 12.0},  {15.0, 0.0, 0.0, 12.0, 0.0},  {16.5, 0.0, 0.0, 14.0, 0.0},  {8.0, 14.0, 0.0, 8.0, 10.0}, //s2  {10.0, 16.0, 0.0, 10.0, 12.0},  {0.0, 0.0, 0.0, 14.0, 0.0},  {0.0, 0.0, 0.0, 16.0, 0.0},  {8.0, 9.0, 10.0, 0.0, 9.0}, //s3  {10.0, 12.0, 14.0, 0.0, 14.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 0.0, 0.0, 0.0},  {0.0, 0.0, 9.0, 10.0, 11.0}, //s4  {0.0, 0.0, 11.0, 12.0, 12.0},  {0.0, 0.0, 14.0, 13.0, 0.0},  {0.0, 0.0, 16.5, 14.5, 0.0}}; | IGNACE, 2 shifts assignés   |  |  |  |  | | --- | --- | --- | --- | | Secouriste | Jour | Debut shift | Fin shift | | 0 | 0 | 8.0 | 10.0 | | 1 | 0 | 10.5 | 13.5 | | 1 | 0 | 15.0 | 13.5 | | 3 | 1 | 9.0 | 12.0 | | 2 | 1 | 14.0 | 16.0 | | 4 | 2 | 9.0 | 11.0 | | 4 | 2 | 14.0 | 16.5 | | 0 | 3 | 8.0 | 11.0 | | 1 | 3 | 12.0 | 14.0 | | 2 | 3 | 14.0 | 14.0 | | 1 | 4 | 10.0 | 12.0 |   STPAUL, 6 shifts assignés   |  |  |  |  | | --- | --- | --- | --- | | # Secouriste | Jour | Début Shift | Fin Shift | | 2 | 0 | 8.0 | 10.0 | | 0 | 1 | 10.0 | 12.0 | | 3 | 2 | 10.0 | 14.0 | | 1 | 3 | 8.0 | 10.0 | | 4 | 3 | 10.0 | 12.0 | | 2 | 4 | 10.0 | 12.0 |   Nb d’heures par secouriste   |  |  |  |  | | --- | --- | --- | --- | | Sec. | Nb H | Sec. | Nb H | | 0 | 7.0 | 3 | 7.0 | | 1 | 7.5 | 4 | 6.5 | | 2 | 6.0 |  | | |

1. Technologies
   1. Technologies à utiliser

Pour la réalisation de notre site internet, nous utilisons principalement html, java Netbeans et PostgreSQL. Le html est la partie avec laquelle l’utilisateur interagi pour pouvoir modifier ou consulter les différentes options de notre plateforme. Par la suite, html communique avec notre base de données qui est créée à partir de PostgreSQL. La gestion des calculs des horaires et de l’inventaire s’effectue à l’aide de java.

* 1. Défis et difficultés
     1. Serveur

Nous avons d’abord pris une mauvaise route quant au serveur. Nous tentions de construire un serveur qui serait accessible sur plusieurs postes et qui permettrait de partager l’information d’un pavillon à l’autre. Toutefois un tel serveur c’est avéré trop complexe à réaliser et nous avons donc recommencé et opté pour un site local pour nous concentrer sur les fonctionnalités du site et non son accessibilité. Nous avons créé une base de données locale accessible sur un poste uniquement au lieu d’un base de données sur un réseau. Nous pourrons tout de même, si nous le désirons rendre le site plus accessible en créant un réseau advenant que nous souhaitions pousser notre projet plus loin.

* + 1. Base de données

La réalisation de la base de données fut problématique puisqu’elle devait être compatible et compréhensible à la fois pour le site en HTML et pour la logistique en Java. Nous avons éprouvé des difficultés lors de la connexion de la base de données au HTML puisque nous utilisions une technologie qui nous était nouvelle. Les données devaient parcourir un chemin complexe pour atteindre le résultat escompté. Tout d’abord entrées manuellement par l’utilisateur sur le site en HTML, les données doivent ensuite être stockées dans un tableau sur la base de données. Après cela, le tableau doit être réorganisé pour être analysé par l’algorithme en java. Une fois les shifts assignés et placés dans une Arraylist, ils sont à nouveau envoyés à la base de données pour être emmagasinés et réorganisés et finalement être transmis à l’utilisateur sur le site en HTML. La base de données a été créé sur Ubuntu et nous avons donc dû apprendre toute une série de commandes spécifiques pour réussir à la faire fonctionner.

* + 1. Html

Nous n’étions pas familiers avec la programmation HTML et l’esthétique d’un site web de qualité. Nous avons donc rencontré des difficultés à bien évaluer l’organisation, les proportions et le placement des éléments adjacents. Nous avons eu a effectuer une recherche exhaustive pour nous faire une idée de quelles étaient nos possibilité concernant le design de notre site web. De plus, la fiabilité et la spécificité des sources nous a mis l’épreuve tout au long de la réalisation du projet.

* + 1. Logique

La logique derrière l’algorithme de générateur d’horaire a été programmé en java et n’a donc pas nécessité l’apprentissage d’un nouveau langage de programmation, toutefois il contenait beaucoup de contraintes uniques puisque son objectif était d’offrir un service qui normalement était effectué par une intelligence humaine et donc capable de jugement.

2.2.5 Temps

La gestion du temps fut problématique tout au long de la réalisation du projet puisque nous avons dès le départ pris un gros retard car nous nous étions lancés dans une mauvaise direction et une grande partie du travail a dû être recommencé alors que plusieurs semaines étaient déjà passées. Nous avons tout de même réussi à bien réorganiser notre échéancier et nous avons terminé à temps un programme effectuant l’essentiel de notre idée de base revisitée.

* + 1. Travail d’équipe

Le travail en équipe nécessita une grande communication puisque chacun des membres travaillait sur des technologies différentes qui devaient finalement se regrouper ensemble pour créer un produit final cohérent.

1. Perspectives

Bien que notre projet n’ait pas abouti à un résultat aussi complet et poussé que ce que nous avions espéré, nous pensons tout de même qu’il pourra être utile aux coordonnateurs à venir. Le générateur d’horaire sort un horaire de base qui sauve une grande partie du travail au coordonnateur, ce dernier n’a qu’a l’optimiser légèrement au besoin par la suite et à le placer dans un document plus esthétique et adéquat pour la distribution. Les pages inventaire, secouristes et information du site fonctionnent suffisamment pour être utilisées et faciliter la tâche des gestionnaires en rassemblant à un seul endroit ce qui était auparavant désorganisé et éparpillé.

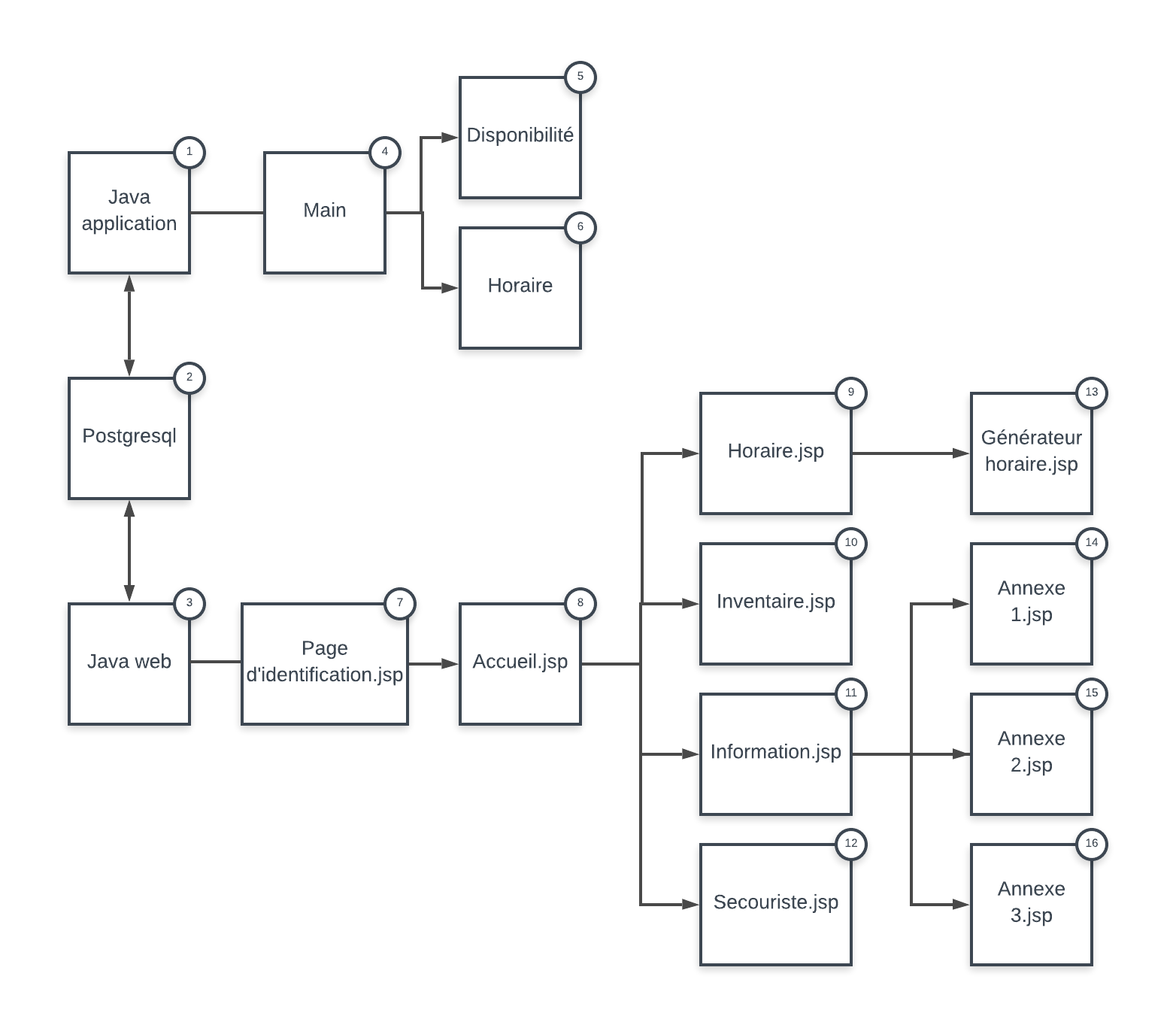
1. Plan
   1. Diagramme de Gantt

|  |  |
| --- | --- |
| |  | | --- | |  | |

* 1. Analyse

|  |
| --- |
| Base de données sur Ubuntu  2  1  4  3  Site en HTML  Algorithme en Java  L’utilisateur rentre des valeurs qui sont transmises et emmagasinées dans la base de données  1  2  La base de données transmet les valeurs au code en java pour qu’elles soient réorganisées et traitées  L’algorithme retourne un horaire sous forme de liste et elles sont emmagasinées dans une base de données  3    4  La base de données retourne les valeurs au site en HTML sur lequel elles seront affichées pour l’utilisateur |

* 1. UML
     1. Schéma



* + 1. Description du schéma

1 Java application :

L’utilisateur rentre manuellement une à une les disponibilités de chacun des secouristes du lundi au vendredi. Lorsque toutes les informations du secouriste sont entrées, il possible d’ajouter un nouveau secouriste ou de directement générer l’horaire avec les secouristes déjà entrés. L’algorithme de génération de l’horaire a été programmé en java sur NetBeans. Il comprend trois classes : Main, Disponibilite et Horaire.

4 Main

La classe main sert essentiellement à créer un nouvel objet horaire et à lancer la méthode de ce dernier qui passe une à une toutes les étapes de l’algorithme.

5 Disponibilite

La classe Disponibilite est un objet possédant cinq caractéristiques et étant utilisé lorsque qu’un shift est assigné et stocké dans une des deux ArrayLists (heuresAssigneesIgnace, heuresAssigneesStPaul).

6 Horaire

La classe horaire contient la majorité de l’algorithme puisque qu’elle contient l’ensemble de la logistique de la génération de l’horaire.  La logique est séparée en quatre méthodes clés.

La première, remplirTabDispo() permet de transformer les valeurs contenues dans la base de donnée en un tableau compatible avec les traitements qui seront effectués sur les disponibilités par la suite.

Ensuite la méthode casPrioritaire() assigne en priorité les heures des secouristes qui ont des disponibilités plus restreintes, c'est-à-dire ceux qui n’ont qu’une seule disponibilité semaine et/ou qui ne cumulent qu’un total inférieur à quatre heures de disponibilités dans la semaine.  Ainsi, ils seront assurés d’avoir leurs heures avant que le reste du programme n’assigne les heures que selon une logique d’optimisation de l’horaire basée sur couvrir le plus d’heures possible.

Une fois les cas prioritaires pris en charge, la méthode remplissageDeBase() est appliquée deux fois, une fois par pavillon. Elle permet de couvrir le maximum de plage horaire de chaque pavillon en évitant tout recoupement entre deux disponibilités de deux secouristes différents. Elle prend également en compte les heures déjà assignées dans casPrioritaire() pour ne pas les réassigner ou assigner un deuxième secouriste alors qu’il y en déjà un dans le local.

Finalement, la méthode remplissageSecondaire() ajoute le plus de shifts possible pour atteindre les capacités maximales des pavillons sans pour autant excéder  le nombre d’heures approprié par bénévole.

D’autres sous méthodes sont également utilisées pour effectuer de plus petites opérations au sein des quatre méthodes principales.

assignerIgnace() assigne le shift au pavillon Ignace-Bourget, incrémente le nombre de secouriste présent dans le pavillon Ignace-Bourget à chaque demi-heure du shift et incrémente le nombre d’heure total assigné au secouriste

assignerStPaul() assigne le shift au pavillon Saint-Paul, incrémente le nombre de secouriste présent dans le pavillon Saint-Paul à chaque demi-heure du shift et incrémente le nombre d’heure total assigné au secouriste

verifierNbSecouriste() s’assure avant que l’on assigne un shift que le nombre de secouriste dans le local ne dépasse pas le maximum. S’il excède le maximum, la méthode raccourcit le shift pour éviter tout recoupement.

calculDemiHeure() transforme une heure du système horaire sur douze heures en un numéro correspondant à la demi-heure de la journée qui y est associée.

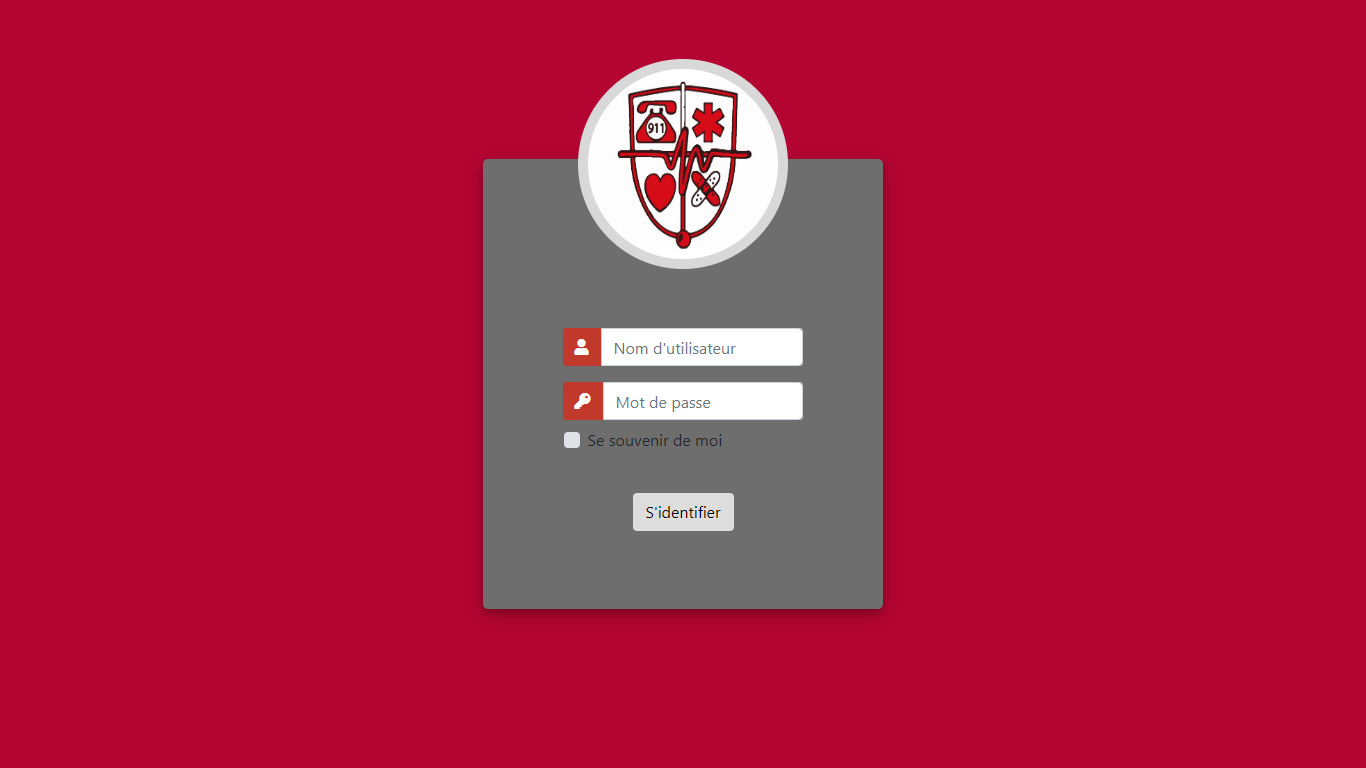
2 PostgreSQL :

PostgreSQL est un langage pour coder une base de données qui en même temps est compatible avec les fichiers .jsp (ex: index1.jsp). L'extension de fichier JSP est liée au format de fichier Java Server Pages et aide à réaliser du contenu personnalisé du côté du serveur sur un site web à l'aide de servlets Java.

3 JavaWeb :

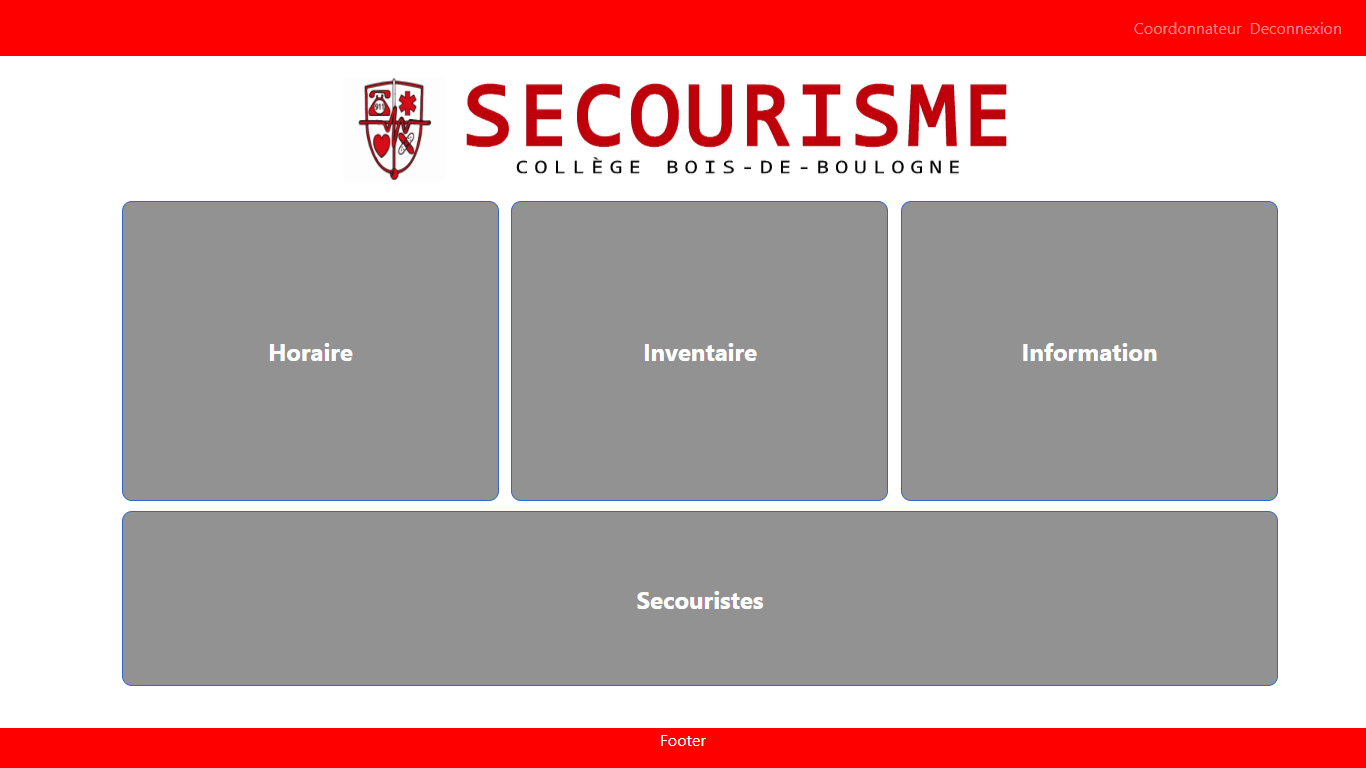
7 Page d’identification.jsp

Lorsque le coordonnateur souhaite accéder au site, il doit entrer son identifiant et son mot de passe. Ces informations sont emmagasinées et cryptés dans notre base de données.



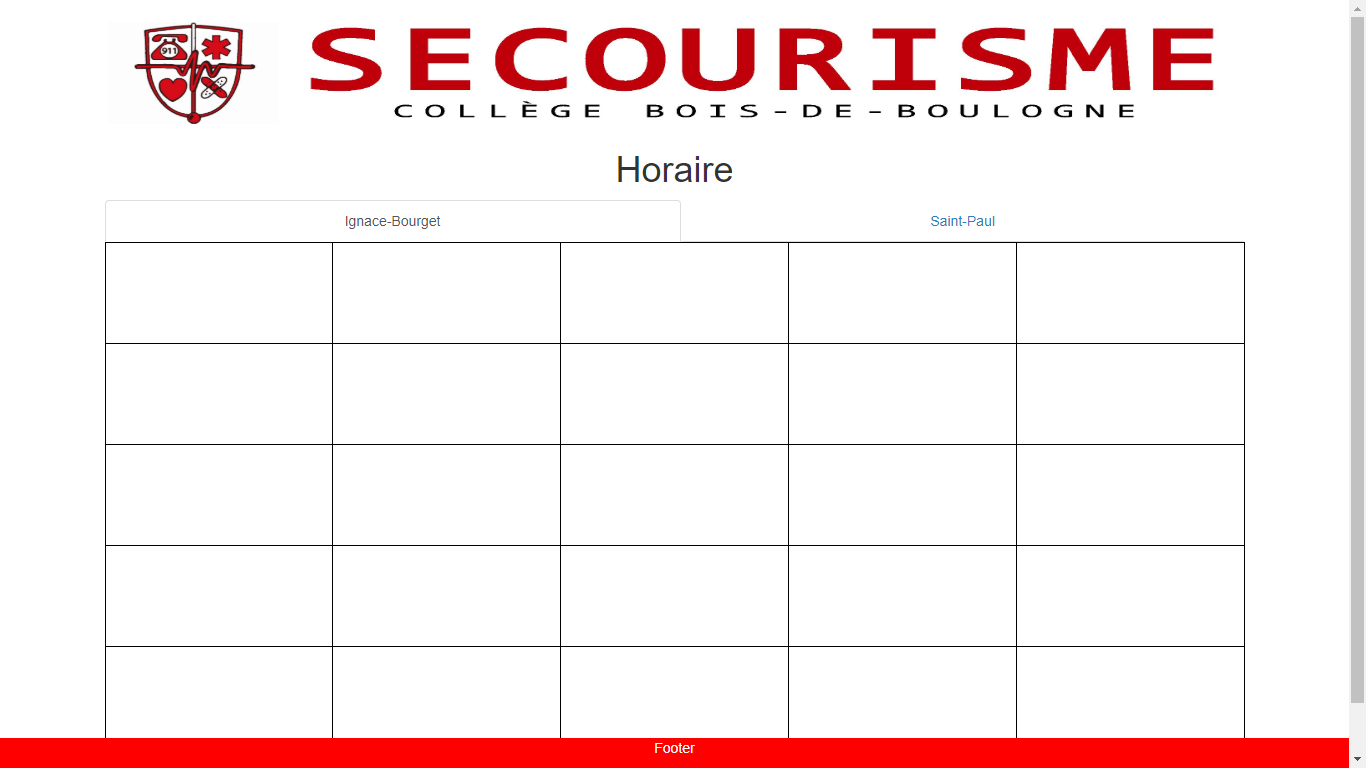
8 Accueil.jsp

La page d’accueil de notre site comprend les quatre options qui s’offrent au coordonnateur ainsi que le logo du club et une option pour se déconnecter. Ici l’utilisateur peut choisir entre Horaire, Inventaire, Information et Secouristes. Ces options seront décrites plus précisément par la suite



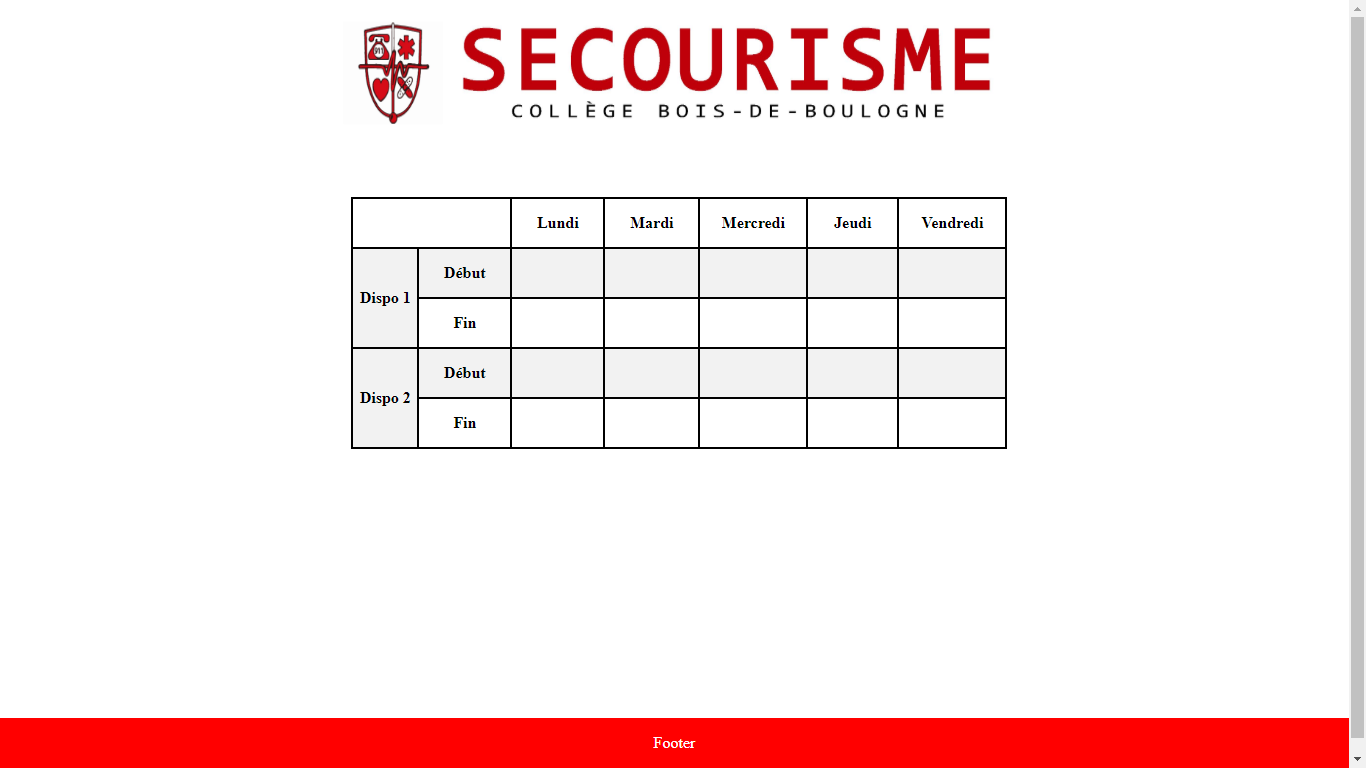
9 Horaire.jsp

Sur la page horaire se trouve le dernier horaire généré par l’algorithme sous forme de tableau à deux onglets, un pour le pavillon Ignace-Bourget, un pour le pavillon Saint-Paul. En dessous se trouve le bouton pour accéder au générateur d’horaire.



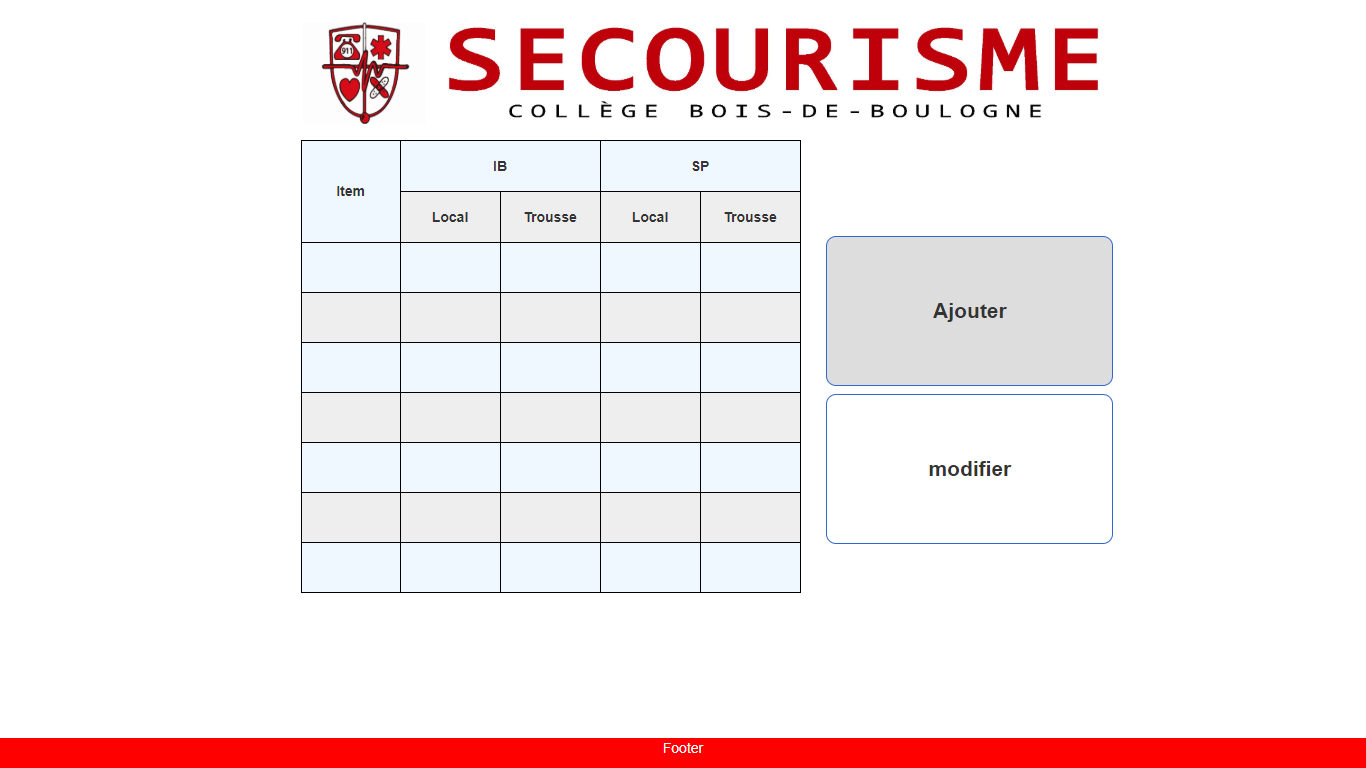
13 Générer Horaire.jsp

À chaque début de session, un horaire devra être généré en accordance avec les disponibilités de chaque secouriste et respectant une série de restrictions précises. Cette fonctionnalité de notre site web permettra de construire automatiquement l’horaire une fois toutes les informations rentrées dans la banque de données. Il doit y avoir un horaire pour chaque pavillon, celui-ci couvre de 8h le matin jusqu'à 16h30 l’après-midi. La priorité est d’assurer, si possible, une présence constante dans le local d’Ignace-Bourget et ensuite de combler le pavillon St-Paul. Une fois toutes les plages horaires couvertes à leur mieux, il est possible d’ajouter des secouristes de surplus dans chaque local pour un maximum de trois secouristes simultanés à Ignace et de deux à St-Paul. Les secouristes doivent se voir attribuer un minimum de trois heures et demie par semaine et un maximum de cinq. Une fois que notre algorithme a généré un horaire adéquat, ce dernier est transformé en un produit visuel attrayant et clair qui peut être partagé avec les secouristes.



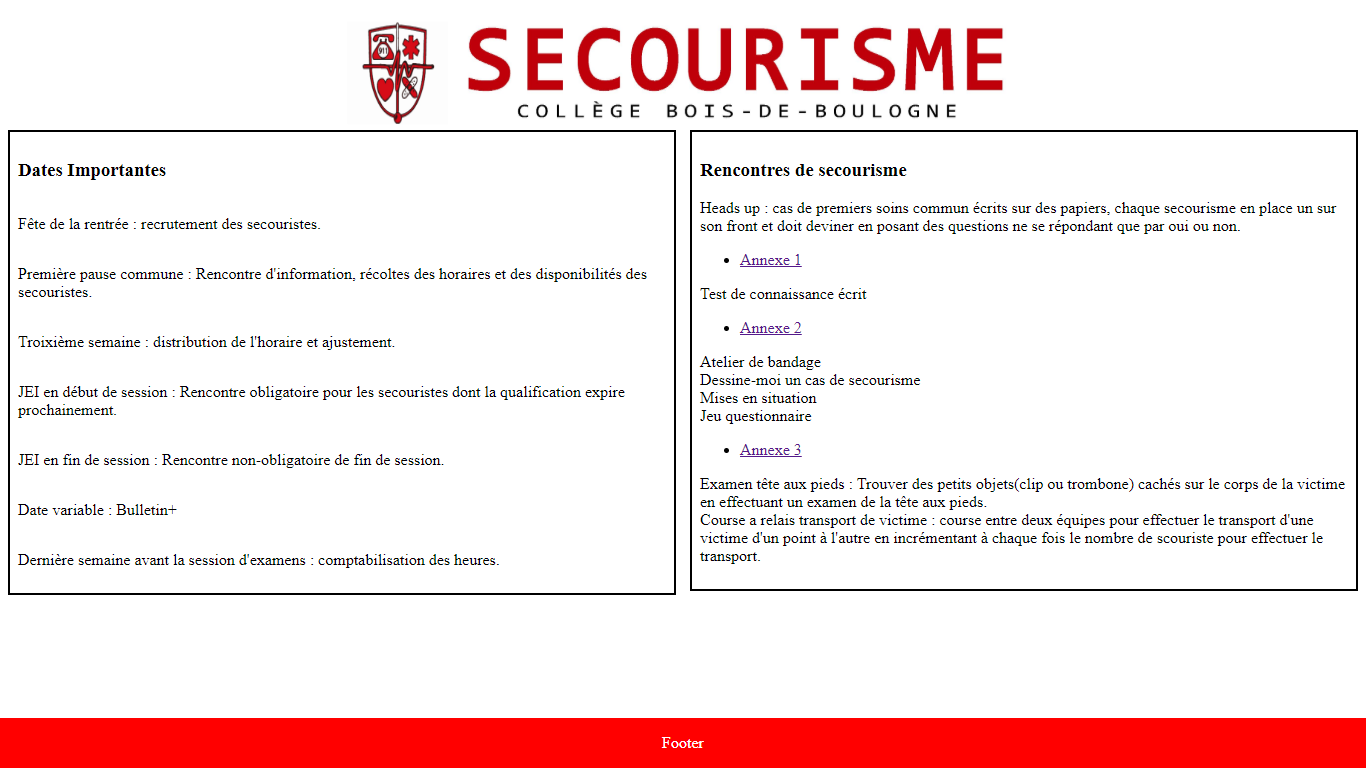
10 Inventaire.jsp

L’utilisateur peut accéder aux inventaires des deux pavillons. Ces inventaires fonctionnent sous un format semblable à un fichier Excel et contiennent les informations pour les locaux et les trousses de premiers soins qui sont utilisées en cas d’interventions d’urgence extérieures. Ces informations sont modifiables par l’utilisateur pour que le coordonnateur puisse modifier l’inventaire en cas de besoin.



11 Information.jsp

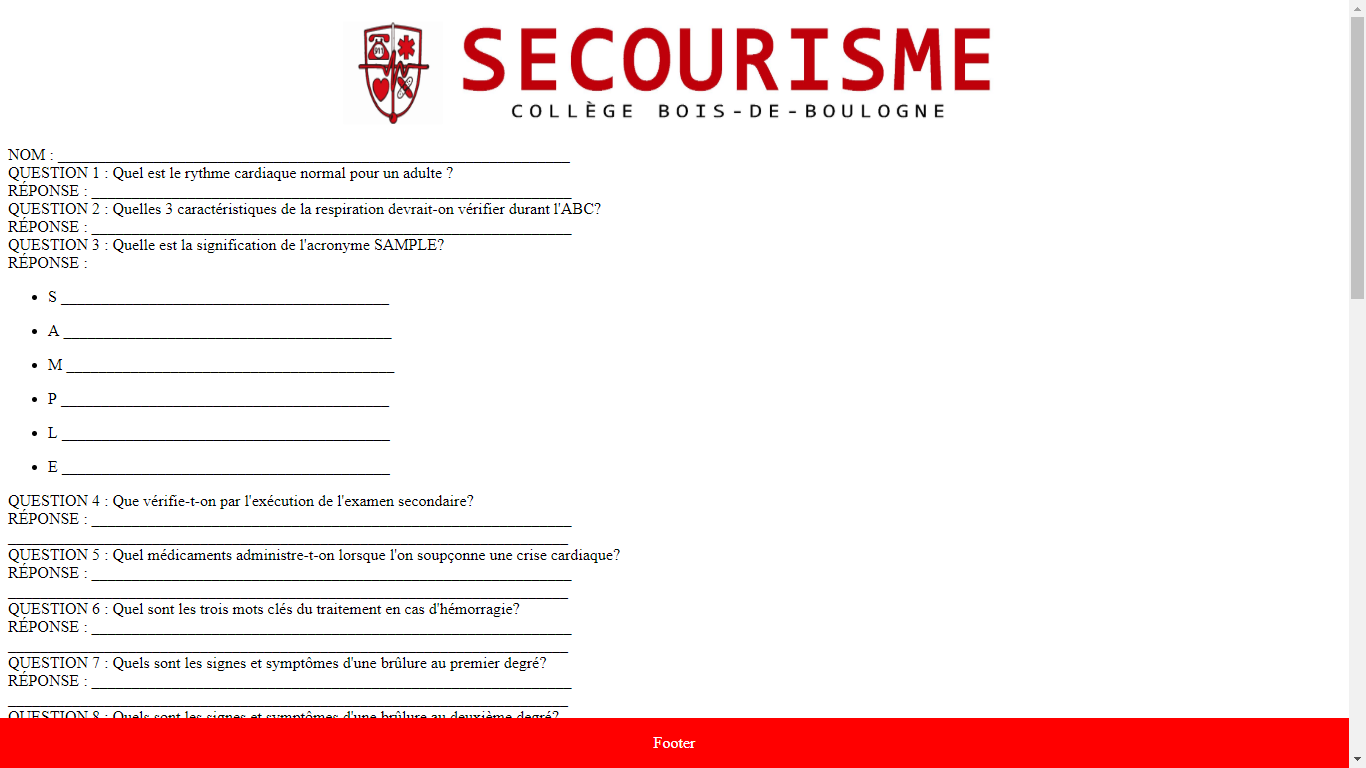
La page d’information contient des idées d’activités pour les rencontres, des notions de secourisme et des informations importantes quant à la gestion et le fonctionnement du club de secourisme. Les coordonnateurs peuvent s’y référer dans le besoin et c’est également un outil de formation pour les nouveaux coordonnateurs lorsqu’ils entrent en fonction.

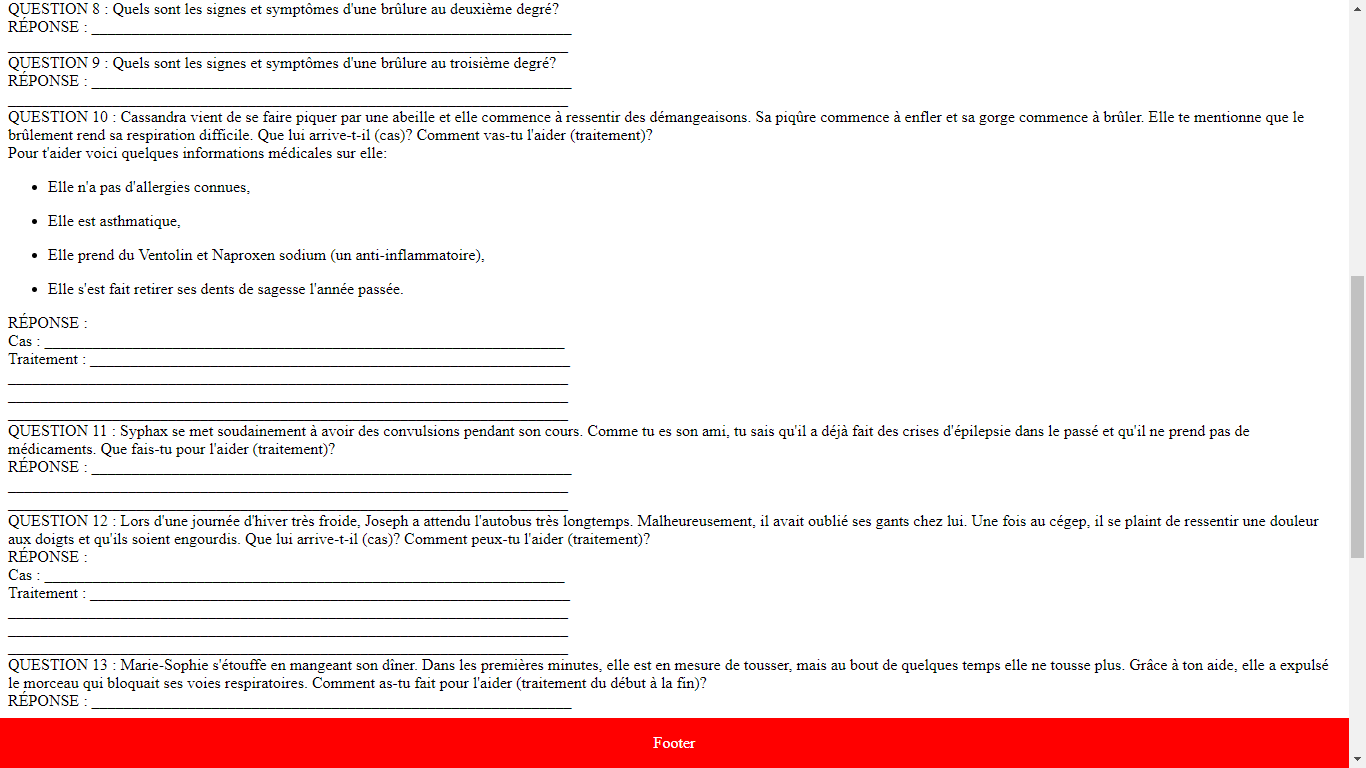


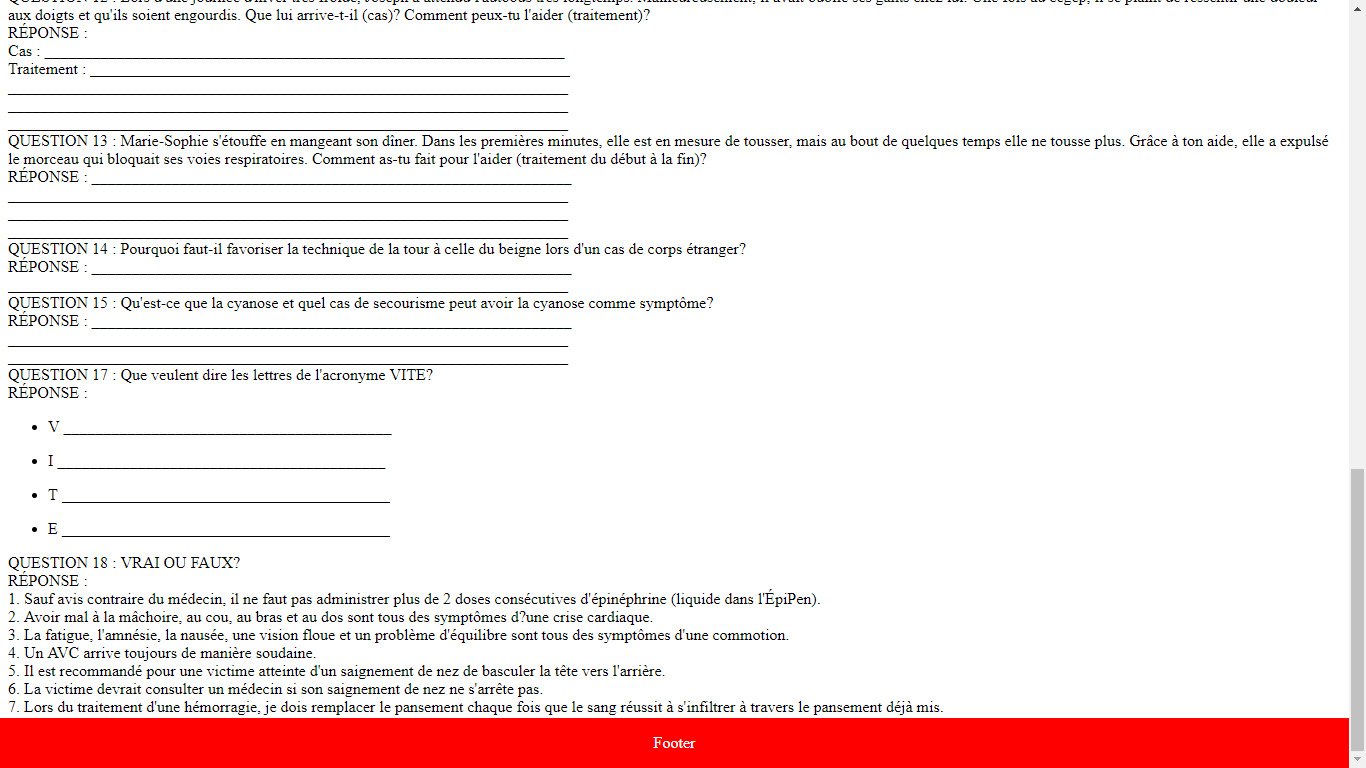
14 Annexe 1.jsp



15 Annexe 2.jsp







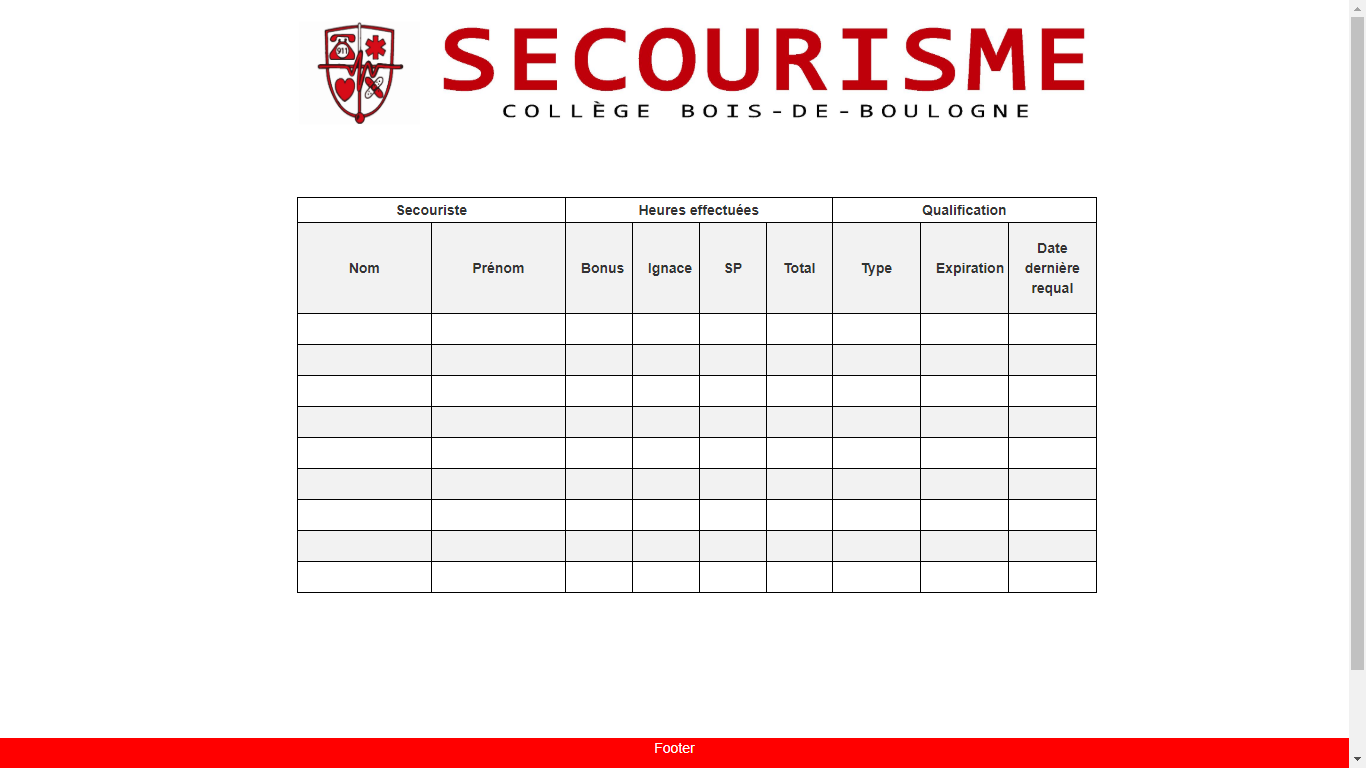
16 Annexe 3.jsp





12 Secouristes.jsp

Cette page contient un tableau affichant les informations pertinentes sur chaque secouriste. Elle permet aux coordonnateurs d’avoir dans un tableau unique toutes les informations concernant chacun des secouristes ainsi que de comptabiliser les heures effectuées.



1. Matériel de référence
   1. Sites de référence

* www.W3SCHOOL.com
* PostgreSQL command line cheat sheet <https://gist.github.com/Kartones/dd3ff5ec5ea238d4c546?fbclid=IwAR2mLRSU_3eJVO6JOZHBq5jN6yFc5_25CIPleXG59eOgo6UO_y1eT3v6Fpg>
* PostgreSQL create table <http://www.postgresqltutorial.com/postgresql-create-table/?fbclid=IwAR1J9PMe1ITKgYwNww5w06LBeG6wfhZTvTq67Wp9dWXFyzpyjiScd-WPrCM>
* How to install PostgreSQL on Ubuntu 14.04 GoDaddy Blog <https://www.godaddy.com/garage/how-to-install-postgresql-on-ubuntu-14-04/?fbclid=IwAR1hKe05rUe-fiQ_DHWNP0AgiPnmfZmQiHvh26INj7OkCeZ8J9ACU43quNU>
* How set connection to PostgreSQL 9.6 database in NetBeans 8 <http://doraprojects.net/blog/?p=3081&fbclid=IwAR0olJTFv3no88u4z6Zli0LpmVzCz4t5REslBIwj86FLPdUa9kcNsFiNUXI>
* Java JDBC PostgreSQL Connection Example <https://examples.javacodegeeks.com/core-java/sql/java-jdbc-postgresql-connection-example/?fbclid=IwAR0VZZcvn6iir9bf46RFOvhVQl4c7aNtcTsIUwNMDaiX9O8QB7r_Q80A9n0>
* Connect java with JavaScript for data visualization in Browser <https://stackoverflow.com/questions/14473268/connect-java-with-javascript-for-data-visualization-in-browser>
* Cours de programmation à Bois-de-Boulogne, introduction à la programmation et programmation orienté objet.

Tutoriels vidéo

* CREER UN SITE ? HTML/CSS #1 - LES BASES <https://www.youtube.com/watch?v=J9w-cir5a6U&list=PLMS9Cy4Enq5JAzNgWPK96HnkE_U7Ol3im>
* How to install NetBeans in Ubuntu <https://www.youtube.com/watch?v=gL7pUleJwiw&feature=youtu.be&fbclid=IwAR0bsxP1XQOFt1cbA2gLzxNMuP37Ov9lraNcXhkaMegzHJlDnUHhcRzXzdc>