

Rapport Projet WEB

Réalisation d'un site web relié à une base de données

Document Utilisateur

Accéder au site

Pour accéder aux différentes fonctionnalités du site, vous pouvez utiliser les différents utilisateurs pré-rentés dans la base de données :

utilisateur@u-psud.fr	(espace utilisateur)
annotateur@u-psud.fr	(espace annotateur, avec des annotations déjà affectées)
validateur@u-psud.fr	(espace validateur)
administrateur@u-psud.fr	(accès à toutes les fonctionnalités)

Mot de passe pour tous les comptes : jambondeparme

Nous conseillons d'utiliser Chromium ou Chrome pour profiter pleinement de l'affichage du site. Firefox gère mal la taille des zones de textes, ce qui peut décaler les lignes.

Page de garde

On accède au site par la page de garde ("page_de_garde.php"). Elle contient un rapide résumé du projet et les noms et prénoms de ses concepteurs. On peut se connecter au site avec son adresse électronique et son mot de passe (bouton Connection), ou créer un nouveau compte (bouton Inscription).

Si l'utilisateur saisit une adresse e-mail qui existe dans la base, mais que le mot de passe ne correspond pas, « Mot de passe invalide » s'affiche. Si l'adresse e-mail n'existe pas, la page affiche « Adresse mail invalide ». Si les identifiants saisis correspondent à ceux entrés dans la base, l'utilisateur est connecté et redirigé vers la page principale (main_utilisateur.php). Les mots de passe sont cryptés.

Inscription

L'utilisateur peut créer un compte en remplissant le formulaire d'inscription. Il doit saisir une adresse électronique, un mot de passe, son adresse, son numéro de téléphone et le rôle désiré. On l'invite à saisir son mot de passe une deuxième fois, afin d'éviter qu'il ne se trompe et qu'il ne puisse plus se connecter. Si toutes les informations sont correctes, l'utilisateur est connecté et redirigé vers la page de garde.

Menu horizontal

L'utilisateur peut naviguer entre les différents espaces grâce au menu horizontal. Les simples utilisateurs n'ont accès qu'à l'espace utilisateur, les annotateurs aux espaces Utilisateur et Annotateur, les Validateurs aux espaces Utilisateurs et Validateurs, et les Administrateurs à

tous les espaces ainsi qu'à l'espace Administrateur. L'affichage des boutons conditionne l'accès de l'utilisateur aux fonctionnalités. Nous avons également ajouté une vérification qui bloque le chargement de la page si l'utilisateur essaye de se connecter directement à une page auquel il ne devrait pas avoir accès. L'utilisateur peut également se déconnecter en cliquant sur Déconnexion, ce qui ferme la session en cours et le redirige vers la page de garde.

Main utilisateur

Tous les types d'utilisateur peuvent accéder à l'espace utilisateur. A l'aide d'un système de formulaire, il est possible d'y interroger base, de sélectionner des gènes / génomes / peptide pour accéder à leurs attributs dans la base, de télécharger les séquences, ou de rechercher les ID dans des bases de données extérieures.

L'espace est séparé en trois menus appelés panneaux.

Menu de recherche

Dans le panneau de gauche, l'utilisateur peut sélectionner à l'aide d'une liste déroulante le type de recherche qu'il souhaite effectuer. Une fois qu'un champ est sélectionné, le formulaire correspondant s'affiche. L'utilisateur peut remplir les champs qu'il souhaite rechercher dans la base de données. Les champs vides ne seront pas considérés. Ce panneau ne se réinitialise que lorsque l'on sélectionne un nouveau type de recherche dans la liste déroulante.

La recherche se fait par recherche de pattern, c'est-à-dire que l'algorithme renvoie le tuple, quelque que soit la place de la chaîne de caractère dans l'attribut de l'objet. Par exemple, si l'utilisateur recherche un gène mais qu'il ne se souvient plus exactement du nom, il peut entrer le début pour sélectionner tous les gènes qui commencent par ces caractères et parcourir les résultats à la main pour retrouver celui qui l'intéresse. Il peut également rechercher des mots clefs dans le champ « Description » pour identifier tous les gènes avec une même fonction métabolique.

L'utilisateur soumet le formulaire en cliquant sur le bouton « Rechercher ».

Panneau des résultats

Le résultat de la recherche s'affiche dans le panneau central. Tous les génomes / gènes / peptides dont les attributs correspondent aux champs saisis sont affichés sous la forme d'un tableau, où chaque ligne correspond à un objet.

A l'extrémité de chaque ligne se trouve un bouton « Voir » qui permet d'afficher les détails de l'objet sélectionné dans le panneau de droite.

Le panneau des résultats ne se réinitialise que lorsque l'on soumet une nouvelle recherche. Néanmoins, chaque fois que la page est rafraîchie (chaque fois que l'utilisateur clique sur un bouton), la page réinterroge la base de données avec les mêmes paramètres de recherche. Pour des recherches courtes, l'affichage est instantané, mais pour des recherches longues (comme sélectionner tous les gènes d'un génome), l'affichage peut prendre plusieurs dizaines de secondes pendant lesquelles l'utilisateur doit attendre.

Panneau de détails

En cliquant sur un bouton Voir, les détails de l'objet s'affichent sur le panneau de droite. L'utilisateur peut accéder aux attributs du génome / gène / peptide sélectionné, ainsi que certains attributs supplémentaires (nom du génome et de l'espèce pour les gènes et les protéines).

Il peut rechercher l'ID de chaque objet dans les bases de données NCBI (génomés, gènes et peptides) ou UniprotKB (gène et peptides). Nous avons choisi ces bases de données pour leur complémentarité. Les bases de données NCBI contiennent beaucoup plus de séquences mais qui ne sont pas forcément annotées, tandis que la base UniprotKB en contient moins, mais toutes annotées et vérifiées à la main.

En cliquant sur le bouton de recherche, l'utilisateur peut ouvrir la séquence peptidique ou nucléotide dans l'outil BLAST du NCBI, choisir les paramètres et le lancer.

Pour les génomes, l'utilisateur peut accéder à un outil de visualisation qui permet de visualiser la séquence à l'intérieur d'une fenêtre définie. Les gènes sont affichés en rouge, avec leurs noms. L'utilisateur peut se déplacer le long de la séquence, ou changer la taille de la fenêtre.

Les recherche d'ID, le BLAST et l'outil de visualisation des génomes s'ouvrent dans une nouvelle fenêtre pour ne pas gêner la navigation dans l'espace de recherche.

Comme les deux autres, ce panneau n'est réactualisé qu'en soumettant une nouvelle requête.

Espace Validateur

Dans cet espace, le validateur peut choisir d'attribuer les annotations. Cet espace est divisé en deux panneaux.

Le panneau de gauche où le validateur peut choisir une séquence parmi toutes celles n'étant pas annotées et un annotateur parmi tous ceux inscrits au moment de l'ouverture de la page. Dans le cas où il n'y a aucune séquence ou annotateur présent, un message indiquant cette absence s'affiche.

Le panneau de droite est lui réservé à la validation des annotations. Le validateur peut choisir une séquence parmi toutes celles en attente de validation. Une fois la séquence choisie, les annotations s'affichent à l'écran et le validateur peut choisir de valider ou refuser les annotations. La base de données se met à jour lorsque l'on clique sur le bouton choisi.

/!\ Par défaut, la base de données ne présente pas de séquence aux annotation en attente de validation. Il faudra passer par un compte annotateur afin de soumettre les premières annotations aux validateurs et à la base de données.

Espace Annotateur

Dans cet espace, l'annotateur peut choisir d'annoter les séquences qui lui ont été attribué. Le panneau de gauche permet de choisir une des séquences à annoter parmi toutes les séquences assignées. Si aucune séquence n'est assignée à l'annotateur, un message adéquat apparait. Enfin, si une unique séquence lui est affecté, elle est soumise d'office à l'annotation.

Le panneau de droite permet à l'annotateur de saisir ces annotations. Le numéro de la séquence lui est rappelé en entête.

Espace administrateur

Cet espace permet à l'administrateur de créer et de supprimer des comptes. Il a également accès aux informations de tous les comptes, notamment à la date de leur dernière connexion. Il n'existe qu'un seul administrateur, et il n'est pas possible de créer un compte administrateur depuis la page web, ni de supprimer un compte administrateur. Pour créer un nouvel administrateur, il faut directement passer par la base de données.

Expérience utilisateur

Au-delà de l'implémentation des différentes fonctionnalités, l'organisation et le design du site ont été pensés pour rendre l'expérience utilisateur la plus agréable possible.

Nous avons fait attention à respecter la taille de l'écran. Toutes les cotes sont relatives : l'utilisateur peut zoomer (dans la limite du raisonnable) sans que les panneaux ne se chevauchent. Tout tient dans une hauteur et une largeur d'écran. Pour accéder aux résultats, l'utilisateur n'a besoin que de déplacer sa souris ou de scroller sur quelques centimètres.

Visuellement, le panneau central est toujours exactement au même endroit quel que soit la page, ce qui fournit un repère à l'œil pour se repérer plus facilement. L'organisation en panneau, plus exigeante au niveau du code, permet de diminuer la distance à parcourir avec la souris à l'intérieur des pages, ce qui accroît le confort de navigation. Le seul outil qui ne respecte pas ces principes est l'outil de visualisation de génomes. A cause de la longueur des séquences, il ne s'intègre pas bien dans le site. Nous avons fait en sorte qu'il s'ouvre dans une fenêtre à part, pour que l'utilisateur puisse l'utiliser avant de revenir sur le site.

Nous avons prêté une attention particulière au rafraichissement des panneaux, notamment pour l'espace utilisateur. Il est très frustrant de devoir cliquer plusieurs fois sur la même liste déroulant pour afficher le même formulaire. Nous avons fait en sorte que la page garde en mémoire le dernier formulaire ouvert quand l'utilisateur clique sur un bouton pour accéder au détail de sa requête. Ainsi, s'il veut rechercher le peptide correspondant à un gène, il peut changer de formulaire et recopier le nom du gène, sans perdre le résultat de sa requête.

Les requêtes SQL sont presque instantanées, ce qui nous permet d'interroger la base régulièrement sans que la page ne mette trop de temps à charger. Néanmoins, le fait de relancer l'affichage à chaque rafraichissement peut poser des problèmes quand la requête renvoi un très grand nombre de résultats qui mettent du temps à s'afficher à l'écran. L'utilisateur doit attendre devant un écran à moitié chargé, ce qui nuit à l'expérience utilisateur. Pire, s'il veut accéder au détail de sa recherche après avoir attendu de longues minutes l'affichage de la page, la page se recharge, et il doit attendre une nouvelle fois. Malheureusement, la manière dont a été pensé le code ne permet pas de corriger ce problème sans revoir toute la structure. De manière générale, la réactivité des pages est excellente et participe à la bonne expérience de l'utilisateur.

Le design est volontairement minimaliste. Nous avons limité l'utilisation d'effets visuels ou d'animation au minimum, à l'exception des boutons, qui changent de couleur et qui modifient la forme de la souris pour que l'utilisateur comprenne immédiatement que ce sont des objets sur lesquels on peut cliquer. Le résultat est très *old school* mais plutôt agréable à l'œil.

Pour les couleurs, nous avons choisi d'utiliser plusieurs nuances de bleu et de jaune. Ce sont deux couleurs complémentaires, très contrastées mais plus douces à l'œil (et plus réjouissantes) que le blanc, le noir et toutes les nuances de gris. Nous aurions pu utiliser des pages de CSS déjà faites, plus complètes, mais nous avons préféré un style dépouillé et fonctionnel.

Conclusion

Le *game feel* est un terme hérité du jeu vidéo qui désigne la sensation tactile intangible que l'on ressent à manipuler un objet numérique, indépendamment de la qualité de ses fonctionnalités.

De la manière qu'il existe un plaisir particulier à interagir avec un jeu vidéo décorrélé des graphismes ou du système de jeu, nous pensons qu'il existe un plaisir unique à naviguer sur une page web. Si l'on remplaçait toutes les données stockées dans notre base par des mots tirés au hasard jusqu'à ce que l'utilisation de l'outil n'ait plus de sens professionnel, nous pensons qu'il subsisterait toujours une forme de satisfaction à naviguer à l'intérieur de nos pages. La plupart des décisions nous avons prises en termes de design, de couleur ou d'organisation participe à améliorer cette expérience utilisateur. Même si certaines fonctionnalités pourraient être ajoutées, nous sommes très contents du résultat, qui correspond à ce que l'on imaginait.

Retour d'expérience : Adrien

Ce projet nous a beaucoup fait progresser en html et en php. On peut s'en rendre compte en comparant la page d'inscription, qui est une des premières pages à avoir été réalisée, avec la page de création d'utilisateur dans l'espace administrateur, qui a été réalisée à la fin. La première nous a pris une après-midi pour un résultat plutôt médiocre, aussi bien niveau du code qu'au niveau visuel. C'était la première fois que nous connectons la page web à la base de données et nous ne maîtrisons ni le système de requête, ni la gestion de l'espace et des classes en HTML et en CSS. La deuxième, pourtant plus complexe, a été réalisée en trois heures, page de style inclus, pour un résultat beaucoup plus probant.

Le manque de temps a conditionné beaucoup de nos choix au niveau du code. C'était la première fois que nous faisons du développement web, et nous avons préféré nous concentrer sur des outils simples comme PHP et HTML. Le résultat est robuste mais pas forcément très élégant et peut s'avérer compliqué à relire.

Nous sommes particulièrement contents de l'espace utilisateur, que ce soit au niveau visuel ou au niveau des fonctionnalités. Nous avons cette idée d'organisation en panneau, que nous avons réussi à implémenter malgré les difficultés. Néanmoins, le code derrière l'espace utilisateur est lourd et souffre de problèmes qui ne peuvent pas être résolus avec les outils que nous avons choisis. Si nous avions été plus à l'aise, nous aurions réalisé la page en javascript avec jquery pour encore améliorer l'expérience utilisateur.

Ombeline

Adrien a endossé plusieurs rôles au cours de ce projet agissant entre autres en tant que développeur et manager.

Pour ma part, le premier contact avec ce projet a été la mise en forme de la base de données. Nous avons d'abord pris le temps de comprendre les données afin d'avoir un ordre d'idée du contenu des annotations, leur type, taille, etc. Un résumé des données possible été obtenu par le script *annotation.py*. Une fois familiarisé avec les données, la structure de la base a été établie. Puis, il nous a fallu formater les données fournies en des fichiers csv afin de pouvoir les intégrer à la base. Le script *formatage.py* est spécialement conçu pour les données fournies et les transformer en format csv adapté exclusivement à notre base de données. Enfin, nous avons enrichi notre base de données en ajoutant des utilisateurs et marquée comme annotation en cours certaines séquences (fichier *dataComp.sql*).

Généralement moins à l'aise qu'Adrien pour fabriquer les pages et le code "*de novo*", j'ai réutilisé et adapté les fonctionnalités d'intérêt pour les pages dédiés aux validateurs et annotateurs.

Documentation Technique

Le site fonctionne en local à partir d'une base de données postgresql local.

Prérequis

Postgresql (de préférence 6.0 ou plus), Apache (PHP) et le module Postgresql pour php (php-pgsql)

Le code est disponible sur github à l'adresse suivante :

<https://github.com/adrienleueur/projetweb.git>

Le repo doit être cloné dans le dossier localhost (var/www/html sous Linux).

Création de la base de données

Pour lancer la base de donnée en local, l'utilisateur doit créer une nouvelle base de donnée nommée projetweb avec la commande `createdb projetweb`, s'y connecter avec la commande `psql -d projetweb` et lancer le fichier databaseCreation.sql dans le dossier DB du repository avec `\i databaseCreation.sql;`

La page web se connecte automatiquement à la base à partir des informations de connexion présentes dans le fichier start.ini (PHP/functions/). Le port utilisé est le port par défaut (5432).

Le script databaseCreation.sql crée un super-utilisateur et lui donne le contrôle complet de la base de données. La page utilise les informations de connexion de ce super utilisateur stockées dans start.ini pour se connecter à la base. Le mot de passe est stocké en clair : théoriquement, cette information est inaccessible à partir de la page, mais nous ne savons pas si cela peut représenter une faille de sécurité.

On accède au site par l'url :

`localhost/projet_web/PHP/page_de_garde.php`

Page de garde

La page de garde est composée d'un titre, d'un texte explicatif, d'un formulaire de connexion et d'un lien vers la page d'inscription.

Le formulaire de connexion effectue une requête sur la table utilisateur, et vérifie s'il existe un tuple correspondant aux informations entrées par l'utilisateur.

Si le résultat n'est pas nul, il crée une session, ce qui permet de sauvegarder des variables superglobales auxquelles on peut accéder tant que la session est ouverte. La page sauvegarde une variable session *connection* (est-ce que l'utilisateur est connecté ?), une variable *email* (email de l'utilisateur connecté) et une variable *role* (rôle de l'utilisateur connecté). La date de dernière connexion est mise à jour dans la table utilisateur à chaque connexion.

Aucun formulaire n'est protégé contre l'injection SQL. Cela peut poser un problème pour le formulaire de connexion, que l'on pourrait utiliser pour accéder aux données personnelles des utilisateurs. C'est une fonctionnalité que nous aurions voulu implémenter. Toutefois, les mots de passes sont stockés sous forme de hash dans la base, ce qui rend leur décryptage théoriquement impossible.

Code commun à toutes les pages (connection.php)

La page connection.php du dossier function est importée au début de chaque page. Elle contient les fonctions qui sont utilisées par toutes les pages :

Les fonctions connect_db() et close_db() permettent respectivement d'ouvrir et de fermer la connexion à la base de donnée. Nous avons vu qu'il était conseillé de l'ouvrir avant et de la fermer après chaque requête plutôt que de la laisser ouverte en continu.

La fonction page_init() est lancée au début de chaque page. Elle vérifie que l'utilisateur est bien connecté, en vérifiant que la variable session CONNECTION est bien initialisée. Si ce n'est pas le cas, la page ferme la session et un script javascript affiche un message d'alerte avant de rediriger l'utilisateur vers la page de garde. Sinon, l'utilisateur peut utiliser la page normalement.

La fonction is_conn_okay() permet de vérifier l'état de la connexion entre la page et la BDD, qui nous a beaucoup servi au début pour faire des tests.

Menu de navigation

Le menu de navigation est présent en haut de toutes les pages. Il est composé d'une liste de boutons-liens, qui permettent naviguer entre les différentes pages. En fonction de la valeur de la variable session 'role', le menu affiche certains boutons. Par exemple, le bouton Espace Validateur s'affiche pour les validateurs et les administrateurs, mais pas pour les simples Utilisateurs et les Annotateurs.

En cliquant sur le bouton Déconnexion, la page ferme la session en cours, supprime toutes les variables de session et redirige vers la page le garde.

Espace utilisateur (main_utilisateur.php)

L'espace utilisateur est la page la plus longue au niveau du code. Elle est organisée en trois volet, qui correspondent aux trois panneaux affichés à l'écran.

Comme nous avons utilisé exclusivement HTML et PHP, nous avons été obligés de tout faire d'un seul bloc.

Plan de la page :

<Affichage du menu de navigation> (l. 16-23)

Affichage du menu de sélection de questionnaire (à gauche) (l. 31-44)

Si génome est sélectionné	Affichage du panneau de recherche de génome (l. 55-77)
	Construction de la query "genome" (180 - 103)
Si gène est sélectionné	Affichage du panneau de recherche de gène (l. 170-163)

	Construction de la query "gene" (l. 177 - 207)
Si prot est sélectionné	Affichage du panneau de recherche de peptides (l. 212-248)
	Construction de la query sur le modèle "prot" (l. 252-280)
Affichage du panneau de résultat (au milieu) (l.305 - 385)	
	Affichage des colonnes en fonction du type de requête (l. 319-340)
	Interrogation de la base de données à partir de la requête (l. 344)
	Affichage des résultats lignes par ligne dans un tableau (l. 346-365)
	Affichage d'un bouton voir à chaque ligne qui renvoie la valeur de la clef primaire de la ligne (l.360)
Affichage du panneau de détail (à droite) (l. 386 - 618)	
Si l'objet à afficher est un génome	Affichage des attributs du génome (l. 400-428)
	Bouton de recherche d'ID (l. 431-435)
	Bouton de téléchargement (l. 436-448)
	Bouton de visualisation (l. 451-458)
Si l'objet à afficher est un gène	Affichage des attributs du gène (l. 462-521)
	Boutons de recherche d'ID (l. 525-533)
	Bouton de téléchargement (l. 541-551)
	Bouton de BLAST (l. 536-552)
Si l'objet à afficher est un peptide	Affichage des attributs du génome (l. 558-613)
	Boutons de recherche d'ID (l. 558-584)
	Bouton de téléchargement (l. 600-608)
	Bouton de BLAST (l. 594-597)

Chaque requête est assemblée en fonction des champs entrés par l'utilisateur dans le panneau à partir d'une base qui dépend du type de recherche. Par exemple, si l'utilisateur sélectionne le menu de recherche de génome, la query est initialisée avec :

```
SELECT cds.nom_cds, gene, gene_symbol, genome.nom_genome
FROM db_genome.cds as cds, db_genome.genome as genome
WHERE cds.nom_genome = genome.nom_genome AND cds.annotateValide = 1
```

S'il rentre un nom de génome dans le formulaire de recherche, la page concatène :

```
" AND nom_genome LIKE '%" . $_POST['q_genome_name'] . "%'" . " ;"
```

à la base, ce qui permet de sélectionner les lignes dont l'attribut nom_genome correspondant à la chaîne de caractère entrée par l'utilisateur.

Comme la page est entièrement réalisée en php, elle est rafraîchie chaque fois que l'on soumet un questionnaire, ce qui réinitialise les variables post des autres questionnaires. Pour améliorer l'expérience utilisateur, nous avons utilisé des variables de session, permanentes, qui définissent quels formulaires et quels panneaux apparaissent à l'écran : last_query, qui gère l'affichage du panneau de gauche, et last_query_output, qui gère l'affichage des panneaux centraux et du panneau de droite.

De manière très simple, les outils de recherche d'ID dans les bases externes insèrent la clef primaire du génome / gène / peptide dans l'URL de la page de recherche de la page web et ouvrent un nouvel onglet vers cette page.

De la même manière, le bouton BLAST insère la séquence à rechercher à l'intérieur de l'URL de la page de requête XML de l'outil BLAST du NCBI, pour que l'utilisateur puisse modifier lui-même les paramètres de la recherche. Le BLAST fonctionne même si la séquence n'est pas clairement affichée sur cette page.

Espace annotateur et Espace validateur

Les espaces annotateurs et validateurs ont été testés pour diverses situations de la base de données (nombre de séquences et d'annotateurs variables).

➤ Espace annotateur

La page contient des requêtes SQL adressée à la base de données. Ici, on veut récupérer les séquences qui sont attribués à l'annotateurs quelque soient leur statut. On veut également récupérer les séquences qu'il lui reste à annoter parmi celle qu'il n'a jamais soumise ou celles dont les annotations ont été refusées.

- `SELECT nom_genome, nom_cds, valide FROM db_genome.attribution_annotateur as anot WHERE anot.mail_annot ='email';`
- `SELECT nom_cds FROM db_genome.attribution_annotateur WHERE mail_annot ='emailSelonAnnotateurConnecté' AND annote=0;`

La page contient un formulaire permettant de choisir les séquences à annoter (une à une) sur une liste déroulante (ou commise d'office si une unique est disponible).

Un tableau permet de visualiser toutes les séquences attribuées à l'annotateur, et ce, quelques soient le statut des annotations. On peut d'ailleurs voir le statut de validation des séquences.

Sous réserve qu'une séquence soit désignée, un nouveau formulaire est présenté. Ce formulaire propose tous les champs à annoter et que l'annotateur peut remplir. En appuyant sur le bouton Annoter, les modifications/annotations sont insérées dans la base de données et l'on modifie les tables *cds* (pour le contenu des annotations) et *attribution_annotateur* (pour le statut de l'annotation) par le biais de nouvelle requête de type UPDATE.

➤ Espace validateur

La page commence par les requêtes SQL propre à la page :

- `SELECT nom_cds FROM db_genome.attribution_annotateur WHERE annote=1 AND valide=0;`
- `SELECT nom_cds FROM db_genome.cds WHERE annoteValide=0 EXCEPT(SELECT nom_cds FROM db_genome.attribution_annotateur);`
- `SELECT email FROM db_genome.utilisateurs WHERE statut='Annotateur';`

Soit respectivement, les séquences que doit reviewer le validateur, les séquences pouvant être attribuée aux annotateurs, les annotateurs existants dans la base.

Un premier formulaire permet de choisir les séquences et les annotateurs au moyen de liste déroulante. Après soumission, la base est modifiée : ajout d'une nouvelle ligne à la table *attribution_annotateur* avec une requete de ce type : `INSERT INTO db_genome.attribution_annotateur VALUES ('nom_genome', 'nom_cds', 'mail_annotateur', 0, 0).`

Un second formulaire permet de reviewer les séquences annotées non validées. Lorsque la séquence est traitée, la liste déroulante s'actualise en se raccourcissant.

Deux boutons sont disponibles : Valider ou Refuser. Le premier modifie d'une part l'attribut *annoteValide* de 0 en 1 dans la table *cds* et d'autre part la valeur de l'attribut *valide* (de 0 en 1) dans la table *attribution_annotateur*. Le bouton Refuser quant à lui passe la valeur de l'attribut *annote* de 1 vers 0. Ainsi, les annotations ne sont pas écrasées mais elles ne sont

pas visible par les utilisateurs (`cds.annotateValide=0`) et l'annotateur ayant soumis les annotations peut renvoyer de nouvelles annotations.

Toutes ces modifications sont réalisées par le biais de requête du type UPDATE.

Espace administrateur

Cet espace permet à l'administrateur de créer et de supprimer des comptes. Il a également accès aux informations de tous les comptes, notamment à la date de leur dernière connexion.

Il est composé de deux panneaux :

- « Créer » contient un formulaire. Un fois le formulaire rempli et soumis, un tuple est inséré dans la table utilisateur avec `pg_query`. Il existe une fonction dédiée, `pg_insert`, mais elle est encore au stade expérimentale et `pg_query` fonctionne très bien avec l'ensemble des commandes postgresql que nous connaissons. La page affiche un message d'alerte pour confirmer la réussite de l'insertion ou message d'erreur.
- « Surveiller » est composé d'une table qui résume les informations des utilisateurs, et d'un formulaire qui permet de supprimer un utilisateur. La table est créée en parcourant le résultat d'un `SELECT *` sur la table utilisateurs. Lorsque l'administrateur supprime un utilisateur, toutes les lignes correspondantes des tables utilisateur et attribution_annotateur sont supprimées.