

SAE2.3 : projet personnel

Important :

Ce projet « perso » est un projet de synthèse des ressources du semestre 2 (et de l'année), permettant une évaluation globale de la compétence RT3 et de ses apprentissages critiques. Il sera utilisé pour la SAE23 mais aussi pour la compétence « programmer » de la SAE24.

Tous les sujets sont différents mais assez proches en termes de structure globale de ce qui doit être réalisé, de sorte que vous pouvez vous aider, malgré la différence des sujets. Par contre soyez conscients que vous serez évalué sur ce que vous avez rendu, et donc devrez effectivement avoir les compétences visées (voir plus bas les méthodes d'évaluation prévues).

Présentation générale du sujet :

Votre but sera de réaliser une petite « appli web » fournissant un « back-end » s'appuyant sur une base de données élémentaire. Votre appli sera constituée de 2 parties : une partie « administration »/statique et une partie « publique »/dynamique, mais il n'est pas demandé de gérer les droits d'accès et les enjeux de sécurité pour la SAE 2.3 (cela pourra être différent pour la SAE2.4).

La partie « administration »/statique permettra essentiellement de fournir une petite interface « CRUD » sur une ou deux de vos tables principales de votre application, et de fournir une interface permettant d'effectuer quelques recherches élémentaires sur des critères à définir dans votre cahier des charges. Cette partie pourra s'inspirer du sujet de TP « gestion étudiant » (les séances 5 « cherrypy »), en rajoutant des affichages sur valeurs extraites d'un formulaire.

La partie « publique » / dynamique permettra de réaliser une (petite), application qui utilisera une ou deux de vos tables principales (voir exemple plus loin). Selon les sujets, cette partie sera soit effectivement une appli pouvant être utilisée par des internautes autres que l'administrateur du site, soit une extension plus perfectionnée de l'interface « administrateur » (et ce sera alors encore une appli « mono-utilisateur »).

Consignes techniques :

- Pour la partie « serveur », vous **devrez** utiliser le framework « CherryPy ».
Attention, l'évaluation terminale du projet à lieu à l'IUT en salles C113A et B, il faut que la configuration « examrt/linux PXE » soit compatible. Il est de votre responsabilité de tester bien avant la date d'évaluation la compatibilité logicielle de la salle, et de demander l'installation potentielle de modules **au moins 30 jours avant l'évaluation pratique, prévue fin mai/ début juin** (et vous devrez vérifier avant l'évaluation !).
- Pour la « persistance », vous utiliserez une base de données relationnelle « MySQL/MariaDB » ou SQLite. **Attention**, il pourra être demandé de changer de SGBDR en cours de projet ou d'évaluation. Pour la partie « administration » de votre projet, votre base devra contenir **au minimum deux tables, liées par une relation de référence**. Vous pourrez par contre choisir de gérer les intégrités référentielles soit à l'aide de votre SGBDR, soit à l'aide de votre code applicatif. Le module python utilisé pour l'accès à une base MySQL sera celui disponible en salles de TP C113 (« **pymysql** »). Votre application « dynamique » devra également utiliser au minimum une autre table, probablement plus ou moins liée aux tables de votre application « statique ».
- La création de la base et insertion d'un « jeu d'essai » (ou valeurs initiales) de la base devront être automatisés, il ne doit pas être nécessaire de passer par une interface graphique interactive (de type « PhpMyAdmin ou SqliteStudio) de façon importante : pour cela vous fournirez lors des premiers livrables un script python capable de créer votre base « vide » (la structure de la base) et (ensuite) de remplir les tables à l'aide de fichiers « csv » (en 2 étapes, éventuellement enchaînées).
- Pour la partie « cliente », vous êtes libres de vos choix techniques, mais l'interface (web) de votre appli devra **au minimum** avoir une page d'accueil « vitrine » (des jolies photos obligatoires). Attention aux accès réseaux lors de l'évaluation (conf « examrt ») et à la taille de vos livrables...

Planning du travail à faire :

Etape 1 : cahier des charges et analyse (15/50)

Vous avez **une ébauche** du sujet que vous avez à réaliser, basée sur les informations que vous m'avez fournies sur vos centres d'intérêts, ou sur ce que j'ai trouvé sur vos sites web du S1. La première étape consiste à vous approprier ce sujet (je ne suis pas un spécialiste de tous les sports ou hobbies possibles) et à reformuler correctement l'application à réaliser, notamment en précisant la partie « administration » et les « cas d'utilisation », et donc à écrire un « cahier des charges » **simple et cohérent**. Vous porterez une attention particulière à bien spécifier les utilisations, les recherches et affichages à effectuer à l'aide de formulaires pour la partie « administration » (attention, il ne s'agit pas d'utiliser uniquement des affichages triés !). De même vous veillerez et à bien définir les cas d'utilisation de la partie « publique ». Pour cette dernière partie, vous pouvez utiliser les notations UML ou décrire (en français) le mode de fonctionnement. Je ne vous fournirai pas l'application dès le début du projet, cette partie pourra faire l'objet de discussion en TP.

Pour le 3 avril 12h dernier délai : A rendre dans mon casier numérique « SAE23_specs » un fichier de 3 à 6 pages max, nommé « *Nom_Prenom_specSAE23.pdf* » (revoir les règles de nommage des fichiers), contenant en première partie le cahier de charges reformulé tel que décrit ci-dessus, et en seconde partie le schéma **conceptuel** de votre base de données.

Les critères d'évaluation de cette partie seront : le respect des consignes, l'appropriation du sujet, la « forme » (présentation, orthographe, ...) du cahier des charges, la pertinence des « critères de recherche et/ou affichages » prévus, des « cas d'utilisation » et le **schéma conceptuel** de la base de données. En outre vous présenterez votre projet en 3 minutes (3 diapos) à votre groupe de TP la semaine du 8 avril.

Etape 2 : réalisation élémentaire en mode CLI (10/50)

Ayant reçu/lu votre cahier des charges, je ferai (peut-être) quelques rectifications sur ce que vous proposez. Vous pouvez néanmoins commencer cette étape avant, voire en même temps que l'étape 1, notamment pour la partie « administration », en vous inspirant du tp « gestion étudiant » !

Le but de cette étape est de (commencer à) créer la partie « statique »/« administration » de votre application en mode CLI (pour commencer la partie « Modèle » ou « accès aux données » de la partie « administrateur »), cf tp « gestion étudiant ».)

Pour le 6 mai 12h dernier délai : A rendre dans mon casier numérique « SAE23_CLI » sous forme d'un répertoire nommé « *Nom_Prenom_SAE23_liv1* », zippé (« *Nom_Prenom_SAE23_liv1.zip* ») :

- un (ou deux) script python « .py » autonome permettant donc de créer la base (vide), et d'importer des données à partir de fichiers « CSV » (voir au-dessus pour le minimum de données à fournir).
- un ou plusieurs fichiers permettant de réaliser au moins une bonne partie des fonctionnalités « CRUD » telles que spécifiées dans votre spec rendue avant, et éventuellement (mais cela n'est pas obligatoire) une ébauche en mode CLI de votre application « dynamique » (cela peut vous permettre de réfléchir à l'interface utilisateur à fournir).
- un fichier de requêtes SQL permettant de créer votre base de donnée pour la partie « statique »/« administration », avec le « jeu d'essai » (effectuez un « dump » de votre base). Le but de la fourniture de ce fichier est de me permettre, si les scripts python de création de la base ne fonctionnent pas, de pouvoir tester tout de même votre application. Il doit contenir la commande de création de la base. Par contre, ce fichier ne doit pas être utilisé par votre application.

Conseil : pour ce livrable, commencez par une version simple de votre application, notamment utilisez des tables avec peu de champs et des jeux d'essais réduits, vous serez toujours ensuite capable de rajouter le reste. Commencez par proposer une interface CRUD simple.

Les critères d'évaluation de cette partie seront : le respect des consignes, la qualité des scripts de création de la base et du jeu d'essai, le code de l'appli en mode CLI, et votre maîtrise de ce code (démontrée en éval pratique).

Le rendu de l'étape 3 (application web complète) sera spécifiée ultérieurement (date de rendu approximative : fin mai-début juin, avec éval pratique.) (25/50)

Un exemple de sujet et sa déclinaison

Ce qui suit est un « exemple » très rudimentaire dont vous pouvez vous inspirer pour spécifier votre projet « perso ». Attention, cet exemple est rédigé de façon schématique, votre « cahier des charges » doit être plus précis en termes d'informations à stocker, en termes d'interfaces utilisateur proposées, etc...

On suppose avoir comme thème une gestion de sportifs (avec leur domicile) et l'appli « dynamique » à réaliser est une petite application de co-voiturage.

Pour la partie « administration / interface CRUD » :

Pour cette partie "back-end" élémentaire : il faut spécifier au moins 1 ou 2 formats d'affichage pertinents et spécifiques à l'application, sachant qu'au moins un affichage sera le résultat d'une « recherche » spécifique (via un formulaire).

Si je prends comme structure de base de données l'exemple de l'application très élémentaire du "carnet d'adresse" qui a servi de base aux premiers TP de Bases de données et qui comportait 2 tables (contact et domicile) :

- il faut bien sûr pouvoir ajouter/supprimer un des « contacts » de ma base de sportifs, et modifier par exemple son adresse/ numéro de téléphone.
- il est clair qu'un format d'affichage évident est la liste des sportifs avec leur adresse (par contre aucun intérêt d'avoir l'affichage juste des domiciles, cf « vues du cours de bases de données).
- on peut proposer un affichage des contacts dont on ne connaît pas l'adresse.
- on peut proposer un affichage de tous les individus qui habitent une ville donnée, cette information étant saisie dans un champ par l'opérateur (exemple d'un formulaire de recherche), ou un affichage des sportifs nés une certaine année.

Pour la partie « publique » :

Pour cette partie « publique » (application de covoit' pour aller à une compétition) : il faut spécifier l'interface d'utilisation (« UI »), ici par exemple :

- Création d'un « évènement » (titre, nom, date)
- Possibilité (par un des sportifs de la base) de proposer des places (nb de places, heure de départ à partir de la ville de domicile du sportif) pour l'évènement.
- Possibilité (par d'autres sportifs) de réserver des places dans un des covoit' proposés.
- Avoir des affichages/traces pertinentes.