

Vade-mecum projet

Votre agenda :

- **Pour le 25 octobre (dernier cours)** : avoir formé le binôme et identifié la base de données utilisée pour le projet.
- **Pour le 1^{er} Bureau d'Étude (9 novembre)** : décrire le projet, formaliser le contrat (cf ci-dessous).
- **De mi-novembre à début février** : définir l'organisation du binôme, développer le Logiciel et préparer une présentation pour la soutenance.
- **8, 14 et 15 février** : soutenir le travail réalisé.

Pour la formalisation du contrat, merci de suivre ce guide !

PARAGRAPHE D'INSCRIPTION

| Nom et Prénom | Classique, CFA, FIE ? | E-mail |
|---------------|-----------------------|--------|
| 1. | | |
| 2. | | |

PARAGRAPHE DE DESCRIPTION

Vous devez maintenant choisir un sujet librement au sein de la thématique proposée.

Thématique : ***manipuler, exploiter, présenter les informations d'une (grosse) base de données publique.***

Autrement dit, il faut choisir des données qui vous intéressent, trouver une base de données, et imaginer ce vous allez en faire dans le cadre d'un Logiciel Python avec un cœur de traitement et une interface graphique.

⇒ En quelques phrases il faut identifier, introduire et décrire le sujet choisi, présenter les enjeux du développement, et définir les états d'achèvement, de performance et de finition visés.

PARAGRAPHE DE SPÉCIFICATION

Vous devez maintenant écrire l'ensemble des exigences permettant de comprendre ce que fera votre Logiciel. A noter que cette spécification est composée de phrases débutant par « Le Logiciel doit... ». A ce stade, il faut vérifier la décomposition en exigences bien formulées, ne préjugant pas de la conception et du codage ultérieurs. Regrouper les exigences par thèmes. Bien préciser les interfaces externes.

| |
|---|
| <i>Exigence n°1 : Le Logiciel doit...</i> |
| <i>Exigence n°2 : Le Logiciel doit...</i> |
| <i>Exigence n°3 : Le Logiciel doit...</i> |
| ... |
| <i>Exigence n°i : Le Logiciel doit...</i> |

PARAGRAPHE DE CONCEPTION

Vous devez maintenant décrire la conception de votre Logiciel. Il faut porter son attention sur la décomposition en modules logiciels autonomes et testables, sur la stricte séparation entre l'interface graphique et le traitement et sur la capacité à répartir le travail de développement entre les deux membres du binôme. Des schémas sont indispensables (diagramme des classes par exemple). Les principes d'architecture doivent être clairement énoncés.

PARAGRAPHE DE TEST

Vous devez maintenant décrire les tests qui permettront de vérifier tout au long du développement que l'avancement est réel et maîtrisé : tout ajout ou changement de code doit donner lieu à un passage des tests pour valider l'absence d'une régression intempestive. Il faut disposer de tests préparés, enregistrés, rejoués et comparés avec un résultat connu à l'avance. Le développement du Logiciel est indissociable de son test.

Merci de préparer en binôme une fiche recto-verso A4 contenant l'ensemble de ces paragraphes rédigés, revus et approuvés par les deux membres du binôme.

*Une première version de cette fiche doit être envoyée aux enseignants (PDF par mail) **avant** le 1^{er} Bureau d'Étude du 9 novembre.*

*Cette version sera revue, corrigée et ajustée pendant le BE, et deviendra le **contrat** du binôme pour la réalisation du Logiciel.*

Bon courage !

Post-scriptum n°1 :

Les jalons suivants sont conseillés et doivent être franchis les uns après les autres :

1. Développer la première version du Logiciel en mode console (pas d'interface graphique).
2. Ajouter les fonctions d'enregistrement et de chargement des configurations du Logiciel.
3. Ajouter les fonctions d'exploitation et de visualisation (à ce stade, un graphique peut être construit puis enregistré en PNG).
4. Enfin, ajouter l'interface graphique et réutiliser les fonctions et classes déjà développées (ne pas oublier d'intégrer des graphiques dynamiques).

Post-scriptum n°2 :

- L'utilisation de bibliothèques Python (paquetages) trouvées sur Internet (par exemple sur PyPI ou GitHub) est autorisée.
- L'utilisation de code source Python trouvé sur Internet est autorisée.
- L'utilisation de code source généré par ChatGPT est autorisée.

(De manière générale, respecter les licences et les droits de propriété intellectuelle. Ne pas oublier de citer les auteurs dans les commentaires de vos fichiers.)

Le Logiciel sera un assemblage de code créé, réutilisé, copié, généré. La maîtrise fine de cet assemblage sera évaluée lors de la soutenance : choix, construction, cohérence, tests.

Post-scriptum n°3 :

Pour trouver une base de données publique :

<https://www.data.gouv.fr/fr/>

<https://ec.europa.eu/eurostat/web/main/home>

<https://data.gov/>

<https://donneespubliques.meteofrance.fr/>

SYNOP : https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=90&id_rubrique=32

<https://www.kaggle.com/datasets>

POUR FINIR, CONTRAINTES IMPOSÉES

- Le projet porte sur l'exploitation d'une base de données.
- Le projet doit utiliser le langage Python, la librairie Pandas pour l'exploitation des données et le framework PyQt pour l'interface graphique.
- Il faut séparer les fonctions/classes de visualisation (interface PyQt, graphiques Matplotlib, print() dans console, etc) et les fonctions/classes de traitements (réalisées avec Pandas, Numpy, etc).
- Il faut écrire des tests dans chaque module (en mode console).

Note : la programmation orientée objet n'est pas obligatoire pour la partie traitement des données.