

NOTICE ELEVE PROJET INFORMATIQUE 2A 2022-2023

Table des matières

Notice élève projet informatique 2A 2022-2023	1
1. Introduction.....	1
2. Organisation	2
a. Les groupes de projet	2
b. Choix du sujet	2
c. Encadrement et suivi d'avancement	3
d. Fiche de temps	3
e. Le chef de projet.....	3
f. Lien avec le module de Compléments d'informatique	4
g. Outils de développement	4
h. Outils de gestion de projet et de communication.....	4
3. Description du travail attendu.....	5
a. Processus de développement.....	5
b. Étude préalable	5
c. Conception générale du logiciel.....	5
d. Réalisation et Validation du logiciel	6
e. Livraison finale collective.....	7
f. Livraison finale individuelle	8
g. La soutenance	9
h. Notation finale.....	9
4. Calendrier	9

1. Introduction

Tous les élèves de deuxième année, répartis en groupes de 4 à 5 élèves, participent à la réalisation du projet informatique. Il permet d'effectuer un approfondissement et une mise en pratique des connaissances acquises lors des enseignements informatiques de 1ère année. La composition des groupes est définie par le département d'enseignement informatique.

Le travail demandé consiste à construire une application permettant de répondre à la problématique du sujet proposé. Ce travail se décompose en 3 grandes phases :

1. Une phase d'étude préalable pour décrire la solution envisagée et planifier les grandes phases de la réalisation (diagramme de cas d'utilisation et diagramme de Gantt) ;
2. Une phase de conception générale de l'application pour décrire les exigences fonctionnelles générales par la modélisation (diagramme d'activité ou d'états, diagramme de classes, modèle de données ...) et planifier la mise en place des fonctionnalités (dépendances, priorités, ...) ;
3. Une phase de réalisation :
 - Mise en place de la base de données ;
 - Développement continu du système en python3 accompagné d'une description graphique du ou des modèles choisis (le modèle d'implémentation) ;

À mi projet, vous livrez le dossier d'analyse correspondant au travail réalisé lors des 2 premières phases. En fin de projet, vous devrez rendre un dossier complet. Vous présenterez vos travaux lors d'une soutenance orale. Les éléments de calendrier page calendrier sont impératifs.

2. Organisation

a. Les groupes de projet

Les groupes ont été composés par le responsable de la matière en tenant compte des niveaux constatés en informatique lors de la première année et en veillant à la répartition des élèves admis sur titre. **La composition des groupes n'est pas modifiable.** Il est essentiel de veiller à la bonne intégration des élèves admis sur titre au sein des groupes.

b. Choix du sujet

Chaque intervenant de projet a proposé un sujet qui devra permettre l'application de l'ensemble des enseignements informatiques dispensés en 1ère année (conception d'applications avec UML, algorithmique, programmation objet avec Python et bases de données relationnelles). Cet intervenant prendra en charge les groupes d'élèves qui traiteront le sujet qu'il a proposé.

Vous devrez choisir votre sujet dans la liste accessible sur la plateforme d'enseignement Moodle, suivant la procédure détaillée ci-dessous :

1. Prenez connaissance des sujets présentés sur la plateforme d'enseignement ;
2. Concertez-vous entre membres d'un même groupe ;
3. Ordonnez les sujets par ordre de préférence sur l'application en ligne accessible via un lien sur le cour Moodle du projet.

Remarque : L'attribution des sujets se fait par une procédure automatique en essayant de respecter au maximum la préférence des groupes. En général, les groupes obtiennent leur premier ou second choix, plus rarement au-delà. Une fois attribué, les sujets ne peuvent pas être modifiés.

c. Encadrement et suivi d'avancement

L'encadrement de l'avancement du projet se déroulera généralement sur 3 heures par semaine. Chaque groupe fera le bilan avec son intervenant du travail qui a été réalisé au cours de la semaine, des difficultés rencontrées et de ce qui doit encore être fait. Afin de formaliser ces échanges, vous pourrez vous appuyer sur une fiche de suivi de séance (voir fiche sur Moodle) ou montrer un kanban correctement rempli.

Avant cet échange, les élèves doivent s'organiser pour :

- Clarifier le bilan des travaux fait dans la semaine (partie « bilan » de la fiche de suivi) ;
- Faire une relecture de code (lors de la phase de développement) ;
- Affecter un numéro de version au code qui servira à l'échange, code qui doit se trouver sur le dépôt git ET être exécutable (lors de la phase de développement).

A chaque séance, l'encadrant notera si un travail sérieux a été réalisé depuis la semaine dernière. Un travail régulier tout le long du projet sera récompenser, tandis qu'un travail fait uniquement avant les rendus sera pénalisé.

Les séances de suivi sont obligatoires. L'appel sera effectué en début et en fin de séance. Votre intervenant pourra demander à tout moment à parler avec la totalité des élèves qu'il encadre.

d. Fiche de temps

Une fiche de temps synthétique disponible sur Moodle est à compléter chaque semaine et vous sera demandée après la soutenance. Cette fiche de temps vise principalement à mieux calibrer la charge du projet informatique d'une année sur l'autre. *Elle n'impacte pas l'évaluation du groupe.* Pour information, le projet informatique vous permet d'obtenir 4 ECTS, ce qui représente un investissement d'environ 100h (séances en présentielle comprises) par élève.

Elle peut aussi vous aider à objectiver l'investissement de chacun dans le projet, mais il faut garder à l'esprit que cette fiche ne permet de visualiser que des apports quantifiables (heures de travail et ligne de code), alors que les apports de chacun peuvent être bien plus divers : animation du groupe, conseils, expertise...

e. Le chef de projet

Au sein de chaque groupe, les élèves devront désigner un chef de projet qui sera responsable de l'animation du groupe et qui organisera les réunions de travail. Il sera également responsable de la répartition et du suivi de l'avancement du travail. Il pourra, pour l'aider dans cette tâche, utiliser la fiche de suivi (voir Moodle). Il devra également veiller au respect du planning général. Il est l'interlocuteur privilégié de l'encadrant pour le groupe qu'il représente. Il doit permettre la discussion et la prise de décision collégiale.

Ce rôle n'a pas être lié aux compétences informatiques, mais à l'appétence pour la gestion d'équipe. Ce rôle pourra être valorisé dans le rapport individuel.

f. Lien avec le module de Compléments d'informatique

Pour vous aider à mener à bien votre projet, un cours de 6h, et quatre TP de 3h sont mis en place dans le module de **Compléments d'informatique**. Lors du cours, vous serez présentés l'organisation générale du projet et quelques concepts nouveaux (systèmes de versionnement (git), appel à une base de données depuis un programme, liens entre données dans un programme et dans une base de données, formats No-SQL, ...). Lors des TP ces concepts seront mis en œuvre.

Bien que ce cours a pour utilité directe de vous aider à mener à bien votre projet, il a pour but final de vous donner des connaissances supplémentaires en informatique pour que vous puissiez évoluer facilement dans le monde de la *data science*.

g. Outils de développement

Les outils de développement à votre disposition seront :

- L'environnement de développement VScode disponible sur votre VM;
- Le logiciel de partage de code et versionnement Git, accessible depuis l'environnement de développement, accompagné d'un dépôt (privé) que vous devrez créer sur un serveur publique (GitLab, Github, Bitbucket) ;
- Une base de données PostgreSQL (hébergé à l'Ensaï) comme système de gestion de base de données (SGBD) pour la persistance des données. Vous utiliserez l'environnement d'un élève sur la base de l'Ensaï pour votre projet ;
- Un certain nombre de librairies et outils pour le lien entre Python et le SGBD, les tests unitaires, la génération automatique de documentations, ...
 - psycopg2 pour se connecter à la base postgresQL, <https://www.psycopg.org/docs/>
 - requests pour réaliser des requêtes http, <https://requests.readthedocs.io/en/master/>
 - unittest pour les tests unitaires, <https://docs.python.org/fr/3/library/unittest.html>
 - pydoc pour la documentation, <https://docs.python.org/fr/3/library/pydoc.html>
 - abc pour les classes abstraites, <https://docs.python.org/fr/3/library/abc.html>
 - flask-restful pour la création d'un webservice REST : <https://flask-restful.readthedocs.io/en/latest/index.html>
 - fastAPI pour la création d'un webservice REST : <https://fastapi.tiangolo.com/>

En plus de cette liste, vous êtes libre d'utiliser les outils de votre choix, *à condition d'être capable de les utiliser sans demander de l'aide inutilement*.

h. Outils de gestion de projet et de communication

Je vous recommande fortement d'utiliser Microsoft Teams pour planifier vos travaux et échanger. Le création d'une conversation pour le groupe me semble le minimal. Mais vous pouvez également installer des plugins pour vous aider à gérer votre projet. Voici quelques propositions

- Trello : pour avoir un kanban collaboratif ;
- Priority Matrix : pour prioriser vos tâches ;

Vous pouvez également utiliser d'autres logiciels pour communiquer, comme Slack ou Discord (qui sont propriétaires comme Microsoft Teams). Je vous déconseille d'utiliser Facebook Messenger ou autre application liée à un réseau social qui ne sont pas utilisées pour gérer un projet en entreprise.

Pour les personnes qui ont du mal à s'organiser mais qui sont sensible à la *gamification* vous pouvez également utiliser Habitica (<https://habitica.com/static/home>) pour vous aider à travailler régulièrement.

3. Description du travail attendu

a. Processus de développement

Le développement s'effectuera collectivement et en cascade. Les phases d'étude préalable, de conception générale, de réalisation et de validation s'enchaînent successivement dans le temps.

b. Étude préalable

Cette 1^{ère} phase d'analyse consiste en la formalisation des besoins décrits par le cahier des charges (le sujet, le point de vue du demandeur ou du maître d'ouvrage). Elle devra permettre d'établir :

- Le périmètre du système d'information cible en précisant les aspects du sujet qui peuvent apparaître flous ;
- Les besoins à satisfaire par le système d'information cible (cas d'utilisation et description des menus) ;
- Les exigences particulières et les contraintes (normes de développement, architecture).

c. Conception générale du logiciel

Cette 2^{ème} phase d'analyse aboutira à un modèle de conception pour répondre aux objectifs fixés par le cahier des charges. Le choix des diagrammes est fait par chaque groupe après validation de leur intervenant. On y trouvera cependant au minimum :

- Un diagramme de cas d'utilisation ;
- Un diagramme de classes ;
- Un modèle physique de données.

En fonction de votre sujet et de ce que vous demande votre intervenant vous pourrez y ajouter :

- Un diagramme de d'activité ou d'état ;
- Un diagramme de séquence ;
- Un modèle logique de données (diagramme entité-relation) ;

Les sujets et les encadrants sont différents, alors préférez faire des choses cohérentes avec ses attendus, que simplement *cocher des cases*.

Si certains diagrammes ne vous disent rien, il y a plusieurs livres sur la modélisation UML à la bibliothèque.

d. Réalisation et Validation du logiciel

La base de données

L'application s'appuiera sur une base de données, que vous devez créer. Pour vous sensibiliser à la variété des données, votre travail devra comporter l'importation ou l'exportation d'un jeu de données. Ce jeu de données sera fourni par l'encadrant dans le format de son choix (XML, JSON, CSV ...).

Codage de l'application

L'application sera réalisée en Python. Le code sera naturellement cohérent avec les diagrammes du modèle de conception déjà établis. Si nécessaire, ces derniers seront mis à jour afin que cette cohérence soit maintenue. **Il n'y a pas d'interface graphique à développer**, la communication avec l'application se faisant par l'intermédiaire de la console. Toutes les fonctionnalités de votre application doivent pouvoir être testées et démontrées via la console.

Validation du logiciel

Tous au long du développement, vous développerez un ensemble de tests :

- Tests unitaires (pour une classe métier au minimum) ;
- Tests utilisateurs pour chaque cas d'utilisation important.

À la différence des autres années, votre tuteur sélectionnera une classe que vous devrez tester entièrement.

Livraison du dossier d'analyse

(Au format électronique uniquement, sur Moodle)

Ce premier livrable est l'aboutissement de la phase d'étude préalable et de conception générale. Il doit montrer que vous avez compris votre sujet et que vous avez une première modélisation de votre application. Ainsi cette livraison contiendra :

- Un planning *détaillé* des 3 phases décrites ci-dessus (diagramme de Gantt incluant un volume horaire prévu pour chaque tâche) ;
- Les diagrammes réalisés pendant la phase d'analyse (voir section [Conception générale du logiciel](#))
- Un document d'architecture spécifiant l'organisation logique des sous-systèmes techniques (paquetages métier, persistance) ;
- ~~Les principaux menus de l'application et leurs enchaînements ;~~
- ~~Une description des fonctionnalités de votre application ;~~
- Une liste des composants à implémenter ainsi que :
 - Leur rôle ;
 - La description de leurs dépendances réciproques ;
 - Le temps de développement prévu pour chacun d'entre eux ;
 - L'ordre de priorité initiale ;

Une taille raisonnable pour ce livrable est de 10 à 15 pages, annexes comprises. Vous pouvez faire plus long mais dans ce cas posez-vous la question : « Ce que j'ajoute est-il nécessaire ? ». Il n'est pas obligatoirement rédigé en LaTeX. *Il sera noté par votre intervenant qui vous en fera un retour.*

e. Livraison finale collective

(Au format électronique uniquement, sur Moodle)

La livraison finale du projet est l'aboutissement de 3 mois de travail et doit décrire et valoriser le travail produit. Elle contiendra au minimum :

- Un rapport de fin de projet qui présentera votre travail. Il devra contenir :
 - **Le contexte de votre application :**
 - **À quels besoins répond-elle ?** Le diagramme de cas d'utilisation n'est pas nécessaire dans le rapport final ;
 - Quels sont les utilisateurs ?
 - L'architecture « macro » de votre application
 - **Est-ce un module python ? Un script sans interaction avec l'utilisateur ? Une application en console ? Une application avec un client et un server qui communiquent via requête http ?**
 - **Les technologies utilisées pour la réaliser**
 - **Un schéma d'architecture (pour faire le lien entre les différents composant de votre application comme votre code, la base de donnée etc)**
 - L'architecture de votre code :
 - **Votre diagramme de classe mise à jour et commenté avec par exemple :**
 - **Explication des héritages/association utilisés**
 - **Présentation des objets métiers et comment ils sont utilisés par l'application**
 - **Mise en avant d'une décomposition en couche, d'un découplage entre les objets.**
 - Le fonctionnement du code :
 - **Zoom sur un processus central en le détaillant complètement. Qu'est-ce que l'utilisateur fait ? Qu'est-ce que le code fait ? Comment interagit-il avec les autre composant ? Quelles sont les classes utilisées ?**
 - Explication *rapide* pour les autres. Votre rapport ne doit pas être un listing des fonctions pythons de votre application.
 - **Présentation du système de stockage et à quels moments il est utilisé dans votre application.** Vous pouvez mettre en avant l'utilisation de forme normale. En effet ce projet doit vous permettre de mettre en avant vos acquis de 1A en la matière.
 - **Les requêtes SQL permettant la création des tables et du jeu de données de test dans un fichier `init_db.sql`. Ces requêtes ne doivent pas figurer dans votre rapport ;**
 - Le code source complet de l'application, dont :
 - Un fichier `main.py` qui permet de lancer votre application ;

- Une classe commentée choisie par votre tuteur correctement ;
- Des tests unitaires pour une classe choisie par votre tuteur ;
- Une fichier README à la racine du projet présentant votre projet. Cette page sera rédigée en anglais et devra :
 - Présenter rapidement votre application (but) avec quelques exemples
 - Expliquer comment l'installer
- Un fichier requirements.txt avec tous les modules pythons à installer pour faire fonctionner votre application.

Les applications les mieux notées pourront être publiées sur le github de l'Ensaï et associé à vos nom (le tout avec votre accord). Mais pour faire cela en plus d'avoir une application de qualité, il vous faudra un README correct.

Bien entendu, **cette liste ne définit que les composantes minimales de votre rendu**. Chaque groupe de projet est libre de fournir toutes les informations complémentaires qui lui paraît pertinentes pour mettre en avant son travail. Cela peut être expliquer des choix de conceptions qui vous semblent pertinents, des outils que vous avez utilisés ou tout autres informations qui vous permet de valoriser votre travail. L'ordre de présentation des différents points, et de leur classement en annexe est libre. **La taille optimale de ce rapport est de 25 pages hors annexe**. Vous pouvez faire plus long mais dans ce cas posez-vous la question : « Ce que j'ajoute est-il nécessaire ? ». Il n'est pas obligatoire de le rédiger en LaTeX.

Pendant le temps entre ce rendu et la soutenance vous avez le droit de modifier votre code, mais ces modifications ne devront être là uniquement pour améliorer la qualité de la démonstration car ce nouveau code ne sera jamais livré pour notation.

f. Livraison finale individuelle

Lors du rendu du rapport chaque élève de chaque groupe joindra une note décrivant son expérience au sein du groupe de projet. Dans cette note d'une à deux pages, vous relaterez aussi bien les bonnes pratiques mises en œuvre que les problèmes rencontrés. Vous pouvez adopter un style moins formel que pour le rapport, mais garder en tête que cette note est lue par le jury au même titre que le reste du rapport. Conserver une mise en page correcte pour cette note. Pour vous aider voici des éléments qui ont leur place dans cette note :

- Votre participation effective au projet (chef de projet, code, animation etc)
- Comment avez-vous vécu le projet avec votre groupe. Est-ce difficile ? Enrichissant ? Eviter les lieux communs, et préférez utilisez des exemples concrets. Vous pouvez mettre en avant des points de tensions qui se sont produits.
- Les enseignements que vous en tirez, ce que vous referiez, ce que vous changerez. Vous pouvez construire cette partie comme une des conseils que vous donnez à vos successeurs.

Nous vous conseillons la relecture de cette production individuelle par les autres membres de votre groupe. Cette note personnelle est très appréciée par les membres du jury, **faites-la sérieusement et honnêtement**.

Vous pouvez vous inspirer de la note individuelle de projet stat de 1A pour rédiger cette fiche

g. La soutenance

Vous présenterez votre travail devant un jury de 3 personnes : un président de jury, qui est un professionnel de l'informatique (du secteur public ou privé), votre encadrant, et un représentant de l'Ensaï. La soutenance consistera en une présentation orale de votre logiciel, partagée équitablement entre les membres du groupe. Elle fera apparaître la méthode de développement utilisée, les choix et options de conception et de programmation retenus. Cette présentation durera 20 minutes, dont 12 minutes environ devront être consacrées à une présentation type diaporama (Beamer ou autre) et 8 minutes à la démonstration de l'application. L'organisation du temps est libre mais doit être dynamique.

Suivra une séance de question du jury d'environ 15 minutes. Puis 5 minutes de retours du jury après délibération.

Quelques conseils pour cette soutenance :

- Présentez la plus-value du groupe (vos différences, vos bonnes idées) ;
- Mettez en avant le projet et pas seulement le logiciel produit (vous n'essayez pas de nous vendre votre application) ;
- Vous pouvez mélanger chronologiquement la présentation et la démo ;
- Illustrez vos transparents avec des schémas / graphiques / captures d'écran / ... ;
- Ne mettez pas de code, mais du pseudo-code (et uniquement si l'algorithme est intéressant).

h. Notation finale

La note finale du projet se répartit comme suit :

- Dossier d'analyse et investissement dans le projet (1/3) ;
- Rapport final et investissement dans le projet (1/3) ;
- Soutenance (1/3).

Le code est noté à part. Cette note est prise en compte par le module Compléments d'informatique.

Si le travail ou l'implication d'un ou plusieurs membres du groupe étaient en deçà des attentes, leur note pourra être dissociée de celle du reste du groupe. **Cela ne sera possible qu'après avoir alerté votre tuteur et le référent de la matière et qu'une discussion avec le groupe n'a eu lieu.** À l'inverse, les membres moteurs du groupe pourront être favorisés.

Vous trouverez sur Moodle les grilles de notation qui sont fournies aux intervenants, pour évaluer votre travail, cette annexe a un rôle de sensibilisation et ne constitue pas un barème définitif.

4. Calendrier

30 août	TP + suivi de projet
9 septembre	TP

16 septembre	TP + suivi de projet
23 septembre	WEI
30 octobre	TP + suivi de projet
8 octobre 23h59	Remise du dossier d'analyse
14 octobre	Suivi de projet
26-28 octobre	3 jours d'immersion avec suivi le premier et dernier jour
18 novembre	Suivi de projet
26 novembre 23h59	Remise du rapport final
06 décembre	Soutenance

Les 26, 27 et 28 octobre auront lieu une période d'immersion. Pendant ses 3 jours vous n'aurez pas cours et pourrez avancer votre projet. Vous verrez votre tuteur le premier pour qu'il vous aide à prioriser vos travaux, et le dernier pour vous vous aider. Une démonstration, même limitée, de votre application le 28 octobre sera valorisée.