

Notice projet informatique de 2^e année

Rémi Pépin

2020-2021

Table des matières

1	Introduction	3
2	Organisation	4
2.1	Les groupes de projet et sujets	4
2.2	Inscription initiale - Adaptation COVID-19	4
2.3	Encadrement et suivi d'avancement	4
2.3.1	Adaptation COVID 19	5
2.4	Fiches de temps	5
2.5	Le chef de projet	6
2.6	Lien avec le module de Compléments d'informatique	6
2.7	Outils de développement	7
2.8	Outils de gestion de projet et de communication	8

3	Description du travail attendu	8
3.1	Processus de développement	8
3.1.1	Étude préalable	8
3.1.2	Conception générale du logiciel	9
3.1.3	Réalisation et Validation du logiciel	9
3.2	Livraison du dossier d'analyse	10
3.3	Livraison finale collective	11
3.4	Livraison finale individuelle	12
3.5	La soutenance	12
3.6	Notation finale	13
3.7	Calendrier	14
A	Fiche de suivi : programmation de compte rendu d'activité hebdomadaire	15
B	Grille d'évaluation du dossier d'analyse et du suivi (première phase)	16
C	Grille d'évaluation du rapport final et de suivi (seconde phase)	17
D	Grille d'évaluation de la soutenance	18
E	Fiche de notation du logiciel produit	19

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 livrerez 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 14 sont impératifs.

2 Organisation

2.1 Les groupes de projet et sujets

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.

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.

2.2 Inscription initiale - Adaptation COVID-19

Historiquement, l'affectation des projets se faisait en fonction de vos vœux après consultation des sujets, mais pour réaliser des emplois du temps plus résilients en cas de confinement partiel, le choix a été fait de vous affecter au hasard un sujet. Ce sujet comme les groupes, ne peut être changé.

2.3 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 (cf. annexe A page 15) 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.

2.3.1 Adaptation COVID 19

En cas reconfinement total ou partiel ces séances pourront être réalisées via l'application Microsoft Teams (ou une autre application proposée par l'encadrant). Une organisation possible est de faire deux tours des groupes. Un premier où les groupes présentent leurs travaux et où l'encadrant propose des pistes d'amélioration, et un second après une petite période de travail (environ 1h30) où les élèves montrent l'évolution suite aux corrections et une planification des prochains travaux. Dans ce contexte et pour profiter pleinement des séances, il est attendu une préparation sérieuse de la part des élèves ainsi qu'un travail en autonomie avec les outils de téléenseignement et de partage mis à leur disposition.

2.4 Fiches de temps

Une fiche de temps synthétique (cf. 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 (séance 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...

2.5 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 (cf. annexe A page 15). 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 liés aux compétences informatiques, mais à appétence pour la gestion d'équipe. Ce rôle pourra être valorisé dans le rapport individuel.

2.6 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 de vous donner des connaissances supplémentaires en informatique pour que

vous puissiez évoluer facilement dans le monde de la *data science*

2.7 Outils de développement

Les outils de développement à votre disposition seront :

- l’environnement de développement Pycharm disponible sur le serveur de calcul de l’Ensaï;
- le logiciel de partage de code et versionnement Git, accessible depuis l’environnement de développement, accompagné d’un dépôt (privé) 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>

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.

2.8 Outils de gestion de projet et de communication

Je vous recommande fortement d'utiliser Microsoft Teams pour planifier vos travaux et échanger. La 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 pour vous aider à travailler régulièrement.

3 Description du travail attendu

3.1 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.

3.1.1 Étude préalable

Cette 1^{re} 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).

3.1.2 Conception générale du logiciel

Cette 2^e 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 séquence;
- un diagramme de d'activité ou d'état;
- un diagramme de classes;
- un modèle logique de données (diagramme entité-relation);
- un modèle physique de données.

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

3.1.3 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 (Modèle de test) 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.

Vous mettrez aussi en place une gestion des erreurs.

3.2 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 3.1.2 page 9);
- un document d'architecture spécifiant l'organisation logique des sous-systèmes techniques (paquetages métier, persistance);
- les principaux menus de l'application et leurs liens;
- une description des fonctionnalités de votre application;
- une liste composants implémentables séparément ainsi que :

- leur rôle;
- la description de leurs dépendances réciproques;
- le temps de développement prévu pour chacun d’entre-eux;
- l’ordre de priorité initial;

Le dossier d’analyse comptera au maximum une quinzaine de page, annexes comprises.

Il sera noté par votre intervenant qui vous en fera un retour.

Il n’est pas obligatoire de le rédiger en LaTeX.

3.3 Livraison finale collective

(au format électronique uniquement, sur Moodle)

La livraison final du projet est l’aboutissement de 3 mois de travail et doit décrire et valoriser le travail produit. Il contiendra au minimum :

- Un rapport de fin de projet qui présentera votre travail. Il contiendra le dossier d’analyse mis à jour des évolutions adoptées lors de la phase de réalisation, notamment la mise en regard des temps de travail initialement prévus et ceux réalisés (dans le diagramme de Gantt et la liste de fonctionnalités), ainsi que la fiche décrite en section 3.4 en annexe;
- les requêtes SQL permettant la création des tables et du jeu de données de test dans un fichier `init_db.sql`;
- le code source complet de l’application, dont :
 - une classe métier correctement commentée;
 - des tests unitaires pour une classe métier;
 - un fichier README à la racine du projet indiquant le chemin vers les deux classes métier précédentes et comment lancer l’application;

Bien entendu, cette liste ne définit que les composantes minimales de votre rendu. Chaque groupe de projet est libre de fournir toute information complémentaire qui lui paraît pertinente pour mettre en avant son travail. L’ordre de présentation des différents points, et de

leur classement en annexe est libre. **Le rapport proprement dit devrait compter au maximum 30 pages** (texte et graphes compris, hors annexes). *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.

3.4 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 de 2 pages, vous relaterez aussi bien les bonnes pratiques mises en œuvre que les problèmes rencontrés. Vous ferez également des propositions pour améliorer le dispositif. 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, faite-la sérieusement et honnêtement.

3.5 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 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).

3.6 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 en annexe 3, 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.

3.7 Calendrier

Voici l'emploi du temps général du projet et des différents rendus attendus :

Choix du sujet : Choix du sujet : Au plus tard le 10 août 12h

7 séances de suivi :

1. 11 septembre 2020
2. 25 septembre 2020
3. 9 octobre 2020
4. 23 octobre 2020
5. 3 novembre 2020
6. 5 novembre 2020
7. 20 novembre 2020

Nommer le chef de groupe : 11 septembre 2020

Séance en autonomie : sur votre emploi du temps

Remise du dossier d'analyse : au plus tard le dimanche 11 octobre à 23h59

Période d'immersion : les 4, 5 et 6 novembre

Remise du rapport final et du code destiné au jury : Au plus tard le vendredi 27 novembre à 23h59

Soutenance : mardi 8 décembre

A Fiche de suivi : programmation de compte rendu d'activité hebdomadaire

Groupe

Tuteur

Semaine

Tâche à réaliser (durée prévue)	Élevés affectés	Bilan (durée réalisée)

B Grille d'évaluation du dossier d'analyse et du suivi (première phase)

Groupe

Tuteur

Relecteur

Critères	Appréciations et commentaires
Dossier d'analyse (15 pts) <ul style="list-style-type: none"> — Contenu <ul style="list-style-type: none"> — Compréhension du problème, planification — Conception et modélisation — Forme <ul style="list-style-type: none"> — Qualité rédactionnelle — Lisibilité générale 	
Suivi (5 pts) <ul style="list-style-type: none"> — Comportement <ul style="list-style-type: none"> — Autonomie des élèves — Aptitude au dialogue et à la communication — Compétence professionnelles <ul style="list-style-type: none"> — Capacité d'organisation — Présentation des résultats — Capacité à progresser 	
Autres remarques	
Appréciation globale et note	

C Grille d'évaluation du rapport final et de suivi (seconde phase)

Groupe

Tuteur

Critères	Appréciations et commentaires
Rapport final (15 pts) <ul style="list-style-type: none"> — Contenu <ul style="list-style-type: none"> — Modèle des besoins, modèle de conception, d'implémentation — Modèle des données, document d'architecture — Forme <ul style="list-style-type: none"> — Qualité rédactionnelle — Lisibilité générale 	
Suivi (5 pts) <ul style="list-style-type: none"> — Comportement <ul style="list-style-type: none"> — Autonomie des élèves — Aptitude au dialogue et à la communication — Compétence professionnelles <ul style="list-style-type: none"> — Capacité d'organisation — Présentation des résultats — Capacité à progresser 	
Autres remarques	
Appréciation globale et note	

D Grille d'évaluation de la soutenance

Groupe

Tuteur

Jury

Critères	Appréciations et commentaires
Structure et contenu de l'exposé - maîtrise du sujet et pertinence de ce qui est présenté	
Forme de la soutenance - maîtrise de l'outil de présentation, expression orale, répartition entre les étudiants, respect du temps imparti	
Démonstration - Exhaustivité fonctionnelle, capacité à ce mettre "au bon niveau"	
Entretien avec le jury - Qualité de l'écoute et de l'argumentation, capacité à prendre du recul	
Autres remarque	
Appréciation globale	
Note	

E Fiche de notation du logiciel produit

Groupe

Tuteur

Critères	Appréciations et commentaires
Qualité de l'application proposée <ul style="list-style-type: none">— Qualité fonctionnelle (validité, robustesse, ergonomie des menus, performance)— Conception : adéquation entre le modèle de conception et le code— Code : structure, normalisation— Maintenabilité : documentation, tests unitaires— Structuration des données— Portabilité : extensibilité, complexité, paramétrage	
Autres remarques	
Appréciation globale	