

## UT'Profiler

Dans ce projet on doit développer l'application UT'Profiler destinée à aider un étudiant d'une **Université de Technologie** à gérer et choisir ses Unités de Valeur (UV) tout au long de son parcours étudiant.

## Fonctions principales

1. Cette application doit permettre de visualiser l'ensemble des UVs existantes. Elle doit aussi permettre l'édition des caractéristiques de ces UVs ainsi que l'ajout ou la suppression d'une UV (en cas de modification de la part de l'université).
2. Cette application doit permettre de visualiser, d'éditer, d'ajouter ou de supprimer les différents types de cursus (Tronc Commun, HuTech, chacune des Branches, PCB d'une branche, PSF d'une filière, mineur).
3. Cette application doit permettre à un étudiant de saisir sa situation actuelle (son dossier) en précisant :
  - dans quels cursus de formation (Tronc Commun, HuTech, une branche, une filière, un mineur, ...) il est inscrit;
  - la liste des UVs auxquelles il a déjà été inscrit avec leur résultat (A, B, C, D, E, F, FX, RES, en cours de réalisation);
  - les éventuelles équivalences de crédits qu'il a obtenu (formation précédente dans un autre établissement, séjour à l'étranger);
  - ...

Un tel dossier pourra être enregistré pour une consultation ou une modification ultérieure.

4. Cette application doit permettre d'aider un étudiant à élaborer la fin de son parcours étudiant grâce à une complétion automatique de ses semestres restants avec des UVs lui permettant de remplir les différents profils liés aux cursus auxquels il est inscrit.

Pour cela, l'étudiant pourra au préalable émettre des exigences, des préférences ou des rejets par rapport à certaines UVs. L'application cherchera alors une bonne solution qui tient compte de ces informations.

L'étudiant pourra aussi entrer des prévisions de semestres d'étude à l'étranger (pour lesquels il doit émettre des bornes inférieures/supérieures sur le nombre de crédits qu'il pense obtenir dans les différentes catégories).

À partir d'une solution proposée par l'application, l'étudiant pourra confirmer (figer) certains choix proposés par l'application, demander d'avancer ou d'en retarder certains, ou au contraire d'en invalider d'autres (rejeter définitivement des UVs). L'application prendra alors en compte ces informations pour calculer une autre solution.

Les solutions calculées pourront être enregistrées pour une consultation ultérieure. Pour un même dossier, il peut y avoir plusieurs solutions enregistrées.

Pour la méthode de complétion automatique, on pourra se contenter d'une méthode simple qui prend au moins en compte les contraintes dures (exigences ou rejets définitifs). La capacité de la méthode à fournir les meilleures solutions ne sera pas évaluée. L'algorithme 1 peut être utilisé. Par contre, la facilité avec laquelle une nouvelle méthode de complétion peut être intégrée sera prise en compte dans l'évaluation.

## Ressources

Vous devrez baser les caractéristiques des différents éléments de votre architecture sur [le guide de l'étudiant](#), qui contient en particulier le règlement des études, et sur [le catalogue des UVs](#) de l'Université de Technologie de Compiègne. Vous pouvez utiliser le fichier `UV_UTC.txt` contenant les caractéristiques de toutes les UVs de l'UTC.

---

**Algorithm 1:** Un algorithme simple de complétion de profil

---

```
Créer une solution qui est une copie du dossier courant;
Trier les UVs selon un ordre qui tient compte des préférences (à adapter en fonction de la façon dont ces
préférences sont renseignées; utiliser une note neutre pour les UVs non évaluées), des inscriptions aux cursus
(certaines UVs deviennent alors obligatoires ou quasi obligatoires), et des rejets (UVs en dernier dans le
classement);
i ← 1;
while toutes les conditions de validation des cursus n'ont pas encore été validées et que toutes les UVs n'ont
pas encore été parcourues do
    if (l'UV i est requise (obligatoire) ou si l'UV i permet d'améliorer le dossier then
        réaliser une inscription à l'UV i le premier semestre possible (au max 35 crédits et 7 UVs pleines par
        semestre et maximum GX09 [avec un objectif GX06]);
        i ← i + 1;
    if toutes les conditions de validation cursus n'ont pas encore été validées then
        informer qu'il n'y a pas de solution compte tenu des cursus choisis par l'étudiant;
```

---

## Livrable attendu

Le livrable est composé des éléments suivants :

- **Code source** : l'ensemble du code source du projet. Attention, ne pas fournir d'exécutable ou de fichier objet.
- **Documentation** : une documentation complète en html générée avec Doxygen.
- **Video de présentation** : une courte video de présentation dans laquelle vous filmerez votre logiciel afin d'en démontrer le bon fonctionnement (maximum 5 min). Pour réaliser cette video, vous pourrez vous servir des logiciels **CamStudio** (Windows), **Jing** (Windows, Mac OS), **RecordMyDesktop** (Linux). Ces logiciels sont mentionnés uniquement à titre d'exemple.
- **Dossier** : votre fichier dossier d'étudiant avec 2 solutions calculées pour ce dossier (qui pourront être utilisés avec l'application).
- **Rapport** : Un rapport en format .pdf composé de 2 parties :
  - la description de votre architecture;
  - une argumentation détaillée où vous montrez que votre architecture permet facilement des évolutions.

Vous pouvez ajouter en annexe de ce rapport des instructions à destination de votre correcteur si nécessaire (présentation des livrables, instructions de compilation, ...). Ce rapport ne devra pas dépasser 10 pages (annexes comprises).

L'ensemble des livrables est **à rendre pour le dimanche 15 juin 23h59 (fuseau horaire de Paris) au plus tard**. Les éléments du livrable doivent être rassemblés dans une archive .zip. L'archive doit être envoyée par mail à votre chargé(e) de TD.

## Évaluation

Le barème de l'évaluation du projet est comme suit :

- **Couverture des fonctionnalités demandées** : 5 points
- **Choix de conception et architecture** : 5 points. En particulier sera évaluée la capacité de l'architecture à s'adapter aux changements. Par exemple, quelles sont les conséquences si on ajoute une nouvelle catégorie d'UV ? Quelles sont les conséquences si l'UT crée des UVs qui donnent des crédits dans plusieurs catégories différentes (par ex, 3 crédits TM + 3 crédits CS) ? Quelles sont les conséquences si la grille de validation du profil TSH est complètement modifiée ? ... De même, on s'attend aussi, par exemple, à ce qu'il soit facile de changer de méthode de complétion automatique des profils ou de stratégie de persistance des données (base de données,...).
- **Evaluation des livrables** : 6 points (code source, documentation, vidéo, exemples de fichiers, rapport)
- **Respect des consignes sur les livrables** : 2 points (échéance, présence de l'ensemble des livrables et respect des consignes sur les livrables).

- **Présence** aux séances de TD : 2 points

Remarque : il est rappelé qu'une note inférieure ou égale à 6/20 au projet est éliminatoire pour l'obtention de l'UV.

## Consignes

- Le projet est à effectuer en binôme (un seul monôme ou trinôme toléré par groupe de TD).
- Vous êtes libre de réutiliser et modifier les classes déjà élaborées en TD pour les adapter à votre architecture.
- En plus des instructions standards du C++, vous pouvez utiliser l'ensemble des bibliothèques standards du C++ et le framework Qt à votre convenance.
- Il n'y a pas de contraintes concernant les éléments d'interface ou sur la gestion/représentation des préférences sur les UVs (soyez créatifs). Il devrait y avoir autant d'interfaces différentes que de binômes.

## Conseils

- La partie difficile du projet est la conception de votre architecture : c'est dessus qu'il faut concentrer vos efforts et passer le plus de temps.
- Il est conseillé d'étudier au moins les design patterns suivants qui pourraient être utiles pour élaborer l'architecture de votre projet : decorator, factory, abstract factory, bridge, composite, iterator, template method, adapter, visitor, strategy, facade, memento. En plus de vous donner des idées de conception, cette étude vous permettra de vous approprier les principaux modèles de conception.
- Vous êtes libre d'élaborer vos formats de fichier. Il est tout de même conseillé d'utiliser XML et d'utiliser les outils XML de Qt.
- Au lieu d'utiliser des fichiers, vous pouvez utiliser un SGBD comme SQLite.