

# Webforce3

## Formation POO en Python

### Projet de fin de séquence

#### Cahier des charges express

##### Objectif

Mister M. voudrait une application stockant les informations sur ses étudiants. Cette application doit permettre aux étudiants (ou à Mister M.) de renseigner ou consulter une fiche étudiant. Mister M. veut par ailleurs pouvoir modifier ou supprimer les fiches étudiants.

**[option]** Dans l'idéal, il aimerait également pouvoir saisir les notes obtenues par ses étudiants et obtenir quelques statistiques à leur sujet.

##### Planning prévisionnel

**Mercredi 16** : Conception de l'appli, maquettage et validation par Mister M.

**Jeudi 17** : Codage de l'appli

**Vendredi 18 à midi** : Livraison de l'appli

##### Spécifications fonctionnelles

Au lancement, l'application doit proposer de s'authentifier comme administrateur (le mot de passe sera "admin"...).

En l'absence de saisie de mot de passe, un simple accès étudiant est donné.

Si la saisie du mot de passe est incorrecte, on peut retenter l'authentification comme administrateur.

Les étudiants doivent pouvoir créer ou consulter une fiche.

Mister M. doit pouvoir créer, consulter, modifier ou supprimer une fiche.

## Contenus de la base de données

Les informations à enregistrer sont :

- les nom et prénom
- l'adresse (numéro, voie, code postal et ville)
- le numéro de téléphone
- le mail
  
- une photo
  
- le niveau estimé dans les différents sujets étudiés : Linux, Algorithmique, UML, HTML, CSS, JavaScript, Bootstrap, Python, Java, SQL (ce niveau sera évalué de 1 à 5 étoiles)

Les données ne seront acceptées par l'appli que si elles sont au format attendu.

Par exemple XX-XX-XX-XX-XX pour un numéro de téléphone ou [xxxxxxxx@xxxxxxxx.xxx](mailto:xxxxxxxx@xxxxxxxx.xxx) pour un mail.

## Spécifications techniques

L'appli doit être codée en Python et posséder une interface graphique construite avec tkinter.

Les données doivent être stockées dans une base SQLite.

L'appli sera développée en DDD (*Doc Driven Development*), en commençant par écrire le mode d'emploi de celle-ci puis la documentation au sein du code (*doctrings*).

Le code sera structuré selon les principes de la Programmation Orientée Objet pour le rendre modulaire et pouvoir le faire évoluer plus aisément (ajout de fonctionnalités futures).

## Critères d'évaluation

| Aspects fonctionnels (9 pts)   | Points max | Evaluation |
|--|------------|------------|
| L'appli fonctionne lorsqu'on la lance  | 1          |            |
| On accède à l'appli directement ou avec un mot de passe pour avoir les droits administrateur     | 1          |            |
| Le mot de passe administrateur est crypté sous la forme d'un <i>hash</i> dans la base de données | 1          |            |
| <b>C</b> - On peut créer une fiche   | 1          |            |
| <b>R</b> - On peut consulter une fiche   | 1          |            |
| <b>U</b> - On peut modifier une fiche si on a les droits administrateur                          | 1          |            |
| <b>D</b> - On peut supprimer une fiche si on a les droits administrateur                         | 1          |            |
| Les entrées non valides de l'utilisateur sont bien gérées  | 1          |            |
| L'appli ne bogue à aucun moment  | 1          |            |
| <i>L'appli a une fonctionnalité supplémentaire</i>   | + 2 pts    |            |
| <b>Design UI / UX (3 pts)</b>  |            |            |
| L'appli a une interface graphique  | 1          |            |
| L'interface utilisateur est agréable à regarder  | 1          |            |
| L'interface utilisateur est agréable à utiliser  | 1          |            |
| <b>Qualité du code (6 pts)</b>   |            |            |
| Code en anglais (interface utilisateur en français)  | 1          |            |
| Nommage pertinent des variables / attributs, fonctions / méthodes et classes                     | 1          |            |
| Les classes sont pertinentes   | 1          |            |
| Les fonctions / méthodes font moins de 20 lignes (1 à 10 lignes dans l'idéal)                    | 1          |            |
| Respect de la PEP 8  | 1          |            |
| Lisibilité globale du code   | 1          |            |
| <b>Documentation du code (2 pts)</b>   |            |            |
| Docstring pour chaque classe et fonction / méthode   | 1          |            |
| Pertinence des docstrings  | 1          |            |