

## Projet de réalisation et jeu de rôles

Votre réalisation s'effectue de la même façon qu'un projet en entreprise. C'est-à-dire que vous devrez vous adresser à différentes personnes occupant des postes clés lors des projets. Ces rôles seront personnifiés par votre formateur. Le formateur endossera les trois (3) rôles suivants : SCRUM master, analyste fonctionnel et développeur en chef. Vous reconnaîtrez le rôle du formateur par le chapeau qu'il portera pendant cette période.

L'objectif de la réalisation est de vous faire baigner dans le même environnement que l'environnement réel. C'est ainsi que ça se passe dans le marché du travail alors c'est ainsi que vos projets doivent être réalisés.

Lorsque le formateur joue un des rôles, vous ne pourrez vous adresser à lui uniquement pour des questions en lien avec ses compétences. Par exemple, si le formateur endosse le rôle analyste fonctionnel et que vous avez un problème technique, vous ne pourrez pas demander l'aide de l'analyste fonctionnel. Vous devrez attendre le retour du développeur en chef, lui envoyer un courriel ou le convoquer en rencontre. Que pourrez-vous avancer pendant que vous attendez l'information? Vous devrez également faire des SCRUM et présenter vos états d'avancement de projet.

### Rôles

Votre rôle : Développeur

Rôle du formateur : Analyste fonctionnel (Chapeau Indiana Jones)

Rôle du formateur : SCRUM Master (Casquette d'armée)

Rôle du formateur : Analyste organique en support aux développeurs (Casquette Batman)

### Méthodologie

Kan ban : Vous devez séparer vos tâches de réalisations. Séparez vos tâches selon votre perception des tâches. Aucune inquiétude, vous pouvez ajuster le tout en cours de route. Pour séparer vos tâches ajoutez chaque tâche sur un post-it et collez le tout dans la section réservé à votre nom. Vous allez ensuite évaluer vos tâches avec le SCRUM Master.

### Dépôt

Votre projet doit être déposé dans *Github* et vous devez inviter le formateur et un collègue de la classe pour tester votre application.

On vous a mandaté pour concevoir du pseudo-code et une application consoles permettant de gérer les animaux d'une clinique vétérinaire. Le patron n'est pas pressé d'obtenir son logiciel car il préfère avoir de la qualité plutôt que de la rapidité (*that is perfect* 😊).

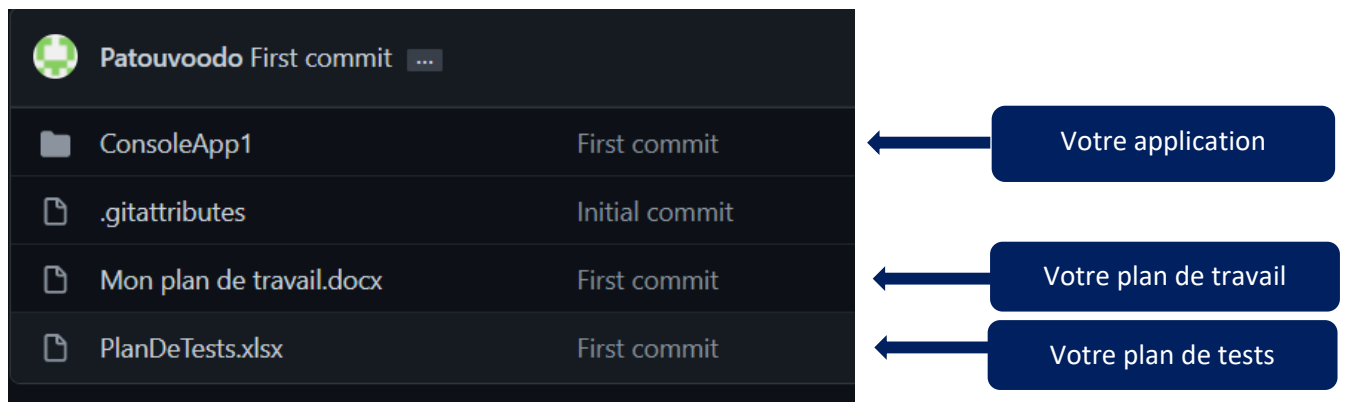
On vous donne un dossier fonctionne avec les spécifications des fonctionnalités. Votre première tâche sera de consulter le dossier et de BIEN SAISIR la demande. Vous avez lu la demande alors relisez la demande une deuxième fois! Si ce n'est pas encore assez clair alors relisez la demande une troisième fois! C'est normal de ne pas comprendre tout du premier coup! Il faut relire plusieurs fois pour s'assurer qu'on n'échappe rien.

Monsieur Lévrier vous présente sa demande dans les pages suivantes. S'il vous manque des détails vous pouvez vous tourner vers l'analyste fonctionnel du dossier.

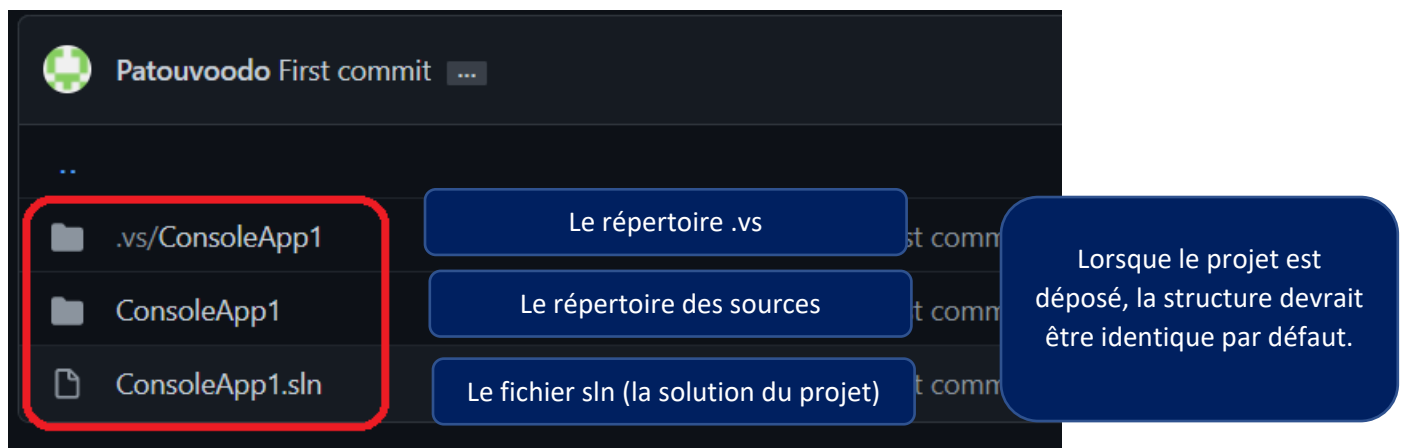
### Ce que vous devez produire :

- 1- Les pseudo-code OU l'organigramme décisionnel de l'application OU un plan de match. Il faut absolument avoir un document qui présente votre stratégie. Il peut prendre la forme que vous voulez mais vous devez faire un plan ou stratégie. Le plan peut tenir sur une seule page!
- 2- Votre application (déposé dans *Github*) et une invitation envoyée au formateur
- 3- Votre plan de tests

Dans Github, votre dépôt doit avoir la **même structure** que celle-ci (les noms peuvent être différents) :



À l'intérieur de votre répertoire de projet, il doit y avoir la structure suivante :



## **Fonctionnalités du logiciel**

Le logiciel doit être en mesure de gérer la liste des animaux en pensions. Le logiciel doit présenter un menu avec les choix suivants :

- 1- Ajouter un animal
  - ID, type d'animal, son nom, son âge, son poids, la couleur (rouge, bleu et violet **uniquement**), le nom du propriétaire
- 2- Voir la liste de tous les animaux en pension
- 3- Voir la liste de tous les propriétaires
- 4- Voir le nombre total d'animaux en pension
- 5- Voir le poids total de tous les animaux en pension
- 6- Voir la liste des animaux d'une couleur (rouge, bleu ou violet)
- 7- Retirer un animal de la liste
- 8- Quitter

Chaque numéro représente un choix du menu. Voici la description de chaque fonctionnalité.

Le menu de démarrage :

Le menu doit présenter les 8 éléments présentés ci-haut. Il doit être disponible tant et aussi longtemps que l'utilisateur ne saisit pas la valeur '8' pour quitter l'application. Tout autre choix ne doit pas être accepté et le système doit gérer la situation.

## 1 - Ajouter un animal

Lorsque l'utilisateur saisie le choix '1', le système doit demander à l'utilisateur de saisir les informations suivantes :

- Type d'animal
- Son nom
- Son âge
- Son poids (des nombres entiers uniquement! Ne vous occupez pas de valider/retirer les points et virgules mais entrez uniquement des valeurs entières positives). Exemple : 12, 20, 58, etc.
- Sa couleur
- Le nom du propriétaire

```
Dans la fonction ajouter un animal
Veuillez saisir le type de l'animal:
chien
Veuillez saisir le nom de l'animal:
winnie
Veuillez saisir l'age de l'animal:
12
Veuillez saisir le poids de l'animal:
35
Veuillez saisir la couleur de l'animal:
rouge
Veuillez saisir le nom du propriétaire de l'animal:
puppy
```

À noter que le système doit conserver en mémoire un ID reliant chaque animal à un identifiant. En d'autres mots, il faut être en mesure de retrouver un animal par l'identifiant.

Le système doit saisir ces informations et les conserver en mémoire tant et aussi longtemps que le logiciel est en exécution. Lorsque le logiciel se termine alors les données ne sont pas conservées.

Lorsque vous ajoutez un animal, l'ajout ne doit pas écraser les données qui sont déjà saisies

## 2- Voir la liste de tous les animaux en pension

Lorsque l'utilisateur saisie la valeur '2' au menu, le système affiche les informations comme le montre l'image présentée ci-bas. Tous les éléments doivent être affichés! **La demande est de présenter les éléments dans le même format. Il peut y avoir des différences entre le nombre d'espace mais la structure doit être la même.**

Exemple :

ID	TYPE ANIMAL	NOM	AGE	POIDS	COULEUR	PROPRIÉTAIRE
0	chien	cookie	12	45	violet	patrick
1	chat	boris	23	654	gris	geneviève

## 3- Voir la liste de tous les propriétaires

Lorsque l'utilisateur saisie la valeur '3' au menu, le système affiche le nom des propriétaires de la façon suivante:

```
-----  
| PROPRIÉTAIRE |  
-----  
patrick  
geneviève
```

#### 4- Voir le nombre total d'animaux en pension

Lorsque l'utilisateur saisie la valeur '4' au menu, le système affiche le nombre total d'animaux comme le montre la figure suivante :

```
-----  
|  NOMBRE ANIMAUX  |  
-----  
2
```

#### 5- Voir le poids total de tous les animaux en pension

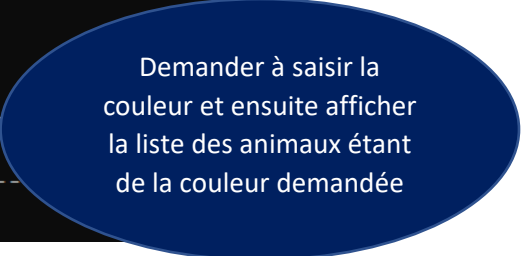
Lorsque l'utilisateur entre le choix numéro '5' au menu, le système doit afficher le poids total de tous les animaux. Voir figure :

```
-----  
|  POIDS TOTAL  |  
-----  
699
```

#### 6- Voir la liste des animaux selon la couleur

Lorsque l'utilisateur saisie une des couleurs (rouge, bleu ou violet), le système doit afficher seulement les animaux ayant la couleur mentionnée. Le système ne doit prendre d'autre couleurs que celles qui sont mentionnées. Dans le contraire, le système redemande de saisir la bonne couleur.

```
VEUILLEZ SAISIR LA COULEUR DE RECHERCHE  
violet  
Dans la fonction voir liste animaux pension  
-----  
ID | TYPE ANIMAL |  NOM  | COULEUR |  
-----  
0  chien      cookie  violet
```



#### 7- Retirer un animal de la liste

Lorsque l'utilisateur désire retirer un animal de la liste, le système doit demander l'identifiant de l'animal à retirer. Lorsque l'animal est retiré, le système affiche la nouvelle liste qui est à jour.

```
VEUILLEZ SAISIR LE ID DE L'ANIMAL
1
Dans la fonction voir liste animaux pension
-----
ID | TYPE ANIMAL |  NOM  | AGE | POIDS | COULEUR | PROPRIÉTAIRE |
-----
0  chien        cookie  23   54   violet  patrick
```

L'utilisateur saisie le ID de l'animal à retirer de la liste. Lorsque l'animal est retiré le système affiche la liste à jour.

#### 8- Quitter

Lorsque l'utilisateur entre la donnée '8' au menu, le système se ferme et l'application est terminée.

#### Règles :

- Lorsque l'utilisateur ne saisit pas un choix valide dans le menu, le système doit afficher le message suivant : **Le choix n'est pas valide...**
- Le nombre de chambres disponibles pour les pensionnaires est limité. Le nombre maximum d'animaux est de 10.
- Lorsque vous ajoutez un animal, l'ajout ne doit pas écraser les données qui sont déjà saisies



**Indice de départ :**

Vous devez avoir une structure pour tenir la liste des éléments.