



## Appartements

Pour ce projet, nous allons repartir du projet existant de l'année dernière. Après avoir pris connaissance de leur projet et de ce qui a été demandé , nous avons fait un petit bilan dans le tableau ci dessous:

<b><i>Fonctions demandées</i></b>	<b><i>Status</i></b>	<b><i>Commentaires</i></b>
<b>Contract 1</b>	Fait	
<b>Contract2</b>	Fait	Refaire StringHelper pour les ToString en contrôlant d'abord les conséquences que cela peut avoir
<b>File1</b>	Fait	ReadAppartement : Refaire la fonction avec de la réflexion
<b>Resources1</b>	Fait	-IconDisplay ne fonctionne pas : en tout cas avec notre configuration
<b>Resources2</b>	Fait	On ne trouve pas le main donc à voir
<b>Interfaces1</b>	Fait	Bien étudier Interfaces 1 et Interfaces 2

<b>Interfaces2</b> <b>Gener</b> <b>Dist</b>		
<b>GUI</b>	Fait	On a encore des erreurs de portabilité avec SWT donc à gérer
<b>ValueDist</b>	Fait	

On remarque que toutes les fonctions demandées ont été faites mais on va en revoir certaines comme dit dans les commentaires.

Notre objectif cette année c'est d'avoir des interfaces pour rendre dynamique les interactions avec l'utilisateur. Pour ce faire nous implémenterons les fonctionnalités suivantes:

1. Lecture et écriture d'un appartement dans et depuis un fichier XML, au format [xmcd-modular](#). Utiliser un élément XML Alternative pour identifier l'appartement, un élément Criterion pour chaque aspect (taille, etc.), et un élément Evaluation pour indiquer les valeurs objectives décrivant l'appartement
2. GUI qui montre à l'utilisateur tous les appartements disponibles dans un répertoire donné en paramètre, triés selon l'utilité de l'utilisateur (fournie dans un fichier dans le même répertoire). L'utilisateur peut cliquer sur un appartement et voit sa description
3. Extraction d'annonces depuis pap.
4. Alertes : l'utilisateur indique à quel niveau d'utilité il veut être alerté en cas d'apparition d'une nouvelle annonce intéressante.
5. L'utilisateur indique son utilité de localisation par rapport à plusieurs lieux.
6. L'utilisateur se fait aider pour déterminer son utilité. Il voit le résultat de sa spécification d'utilité sur des annonces concrètes du système.
7. L'utilisateur peut indiquer en voyant la liste d'annonces qu'il préfère une annonce à une autre alors qu'elle est classé sous la deuxième. Le système lui permet alors de corriger son utilité pour que le classement en question soit

rétabli.

Pour **cette première itération**, notre but est de partir d'un projet tout propre, lisible et compréhensible par tous. On s'est réparti les tâches comme ceci:

- 1er duo : fait le ménage (mise à jour des dépendances, nettoyage dépôt, configuration pom, enlever fichiers inutiles, améliorer la lisibilité du code, faire en sorte de ne plus avoir de warnings)
- )
- 2em duo : change les fonctionnalités mal faites par le groupe de l'année dernière (voir tableau récapitulatif)
- 3em duo: commence la première fonctionnalité de XMCD

Ensuite pour **la deuxième itération**, l'objectif est de terminer les 2 premières fonctionnalités et au moins commencer la troisième. Et comme pour la première itération on aura 3 duo pour chaque fonctionnalité vu qu'ils sont assez indépendants. A priori il faudrait d'abord bien maîtriser tout ce qui est XMCD.

Pour **la troisième itération**, on aimerait finir les 4 dernières itérations restantes concernant les fonctionnalités de l'utilisateur. Il faudrait dans ce cas terminer nécessairement les 3 premières fonctionnalités pour pouvoir faire les tests.

Et pour **la dernière itération**, on va améliorer toutes ces fonctionnalités et/ou ajouter d'autres en fonction de ce qui existe déjà.

A chaque itération, il se peut qu'on ait des tâches de documentation ou de tests unitaires à faire pour compléter les fonctionnalités faites lors des précédentes itérations; on le précisera à chaque début d'itération