

Tests de projet de WebRapport

V1.0

Introduction	2
Test de la partie Utilisateur	2
Test de la partie Projet	7
Test de base donnée	10

Introduction

Ce document est composé de plusieurs tests qui visent à prouver la conformité du produit du projet WebReport. Chaque test permet de s'assurer du bon fonctionnement de chaque fonctionnalités qui composent le projet.

Test de la partie Utilisateur

- Vérifier les attributs des l'utilisateur sont bien définies, vérifier que les données sont bien créées dans la base des données
 - Nom:
 - Prénom:
 - Matricule : 1 lettre + 5 chiffres => déjà attribué par l'admin
 - Mot de passe => 1 généré par défaut, l'utilisateur change à la 1ère co.
 - Métier => Superviseur doit pouvoir ajouter de nouveau métier.
 - Rôles:
 - Statuts => Employé ou ex-employé
 - CI (une attribut qui permet associée les utilisateurs et des activités dans la base de donnée)
- Vérifier que les méthodes sont bien définies et qu'elles sont bien fonctionnées.
 - Superviseur possède toutes les méthodes suivantes, nous devrions vérifier que toutes les méthodes ne peuvent pas être appelées seulement par superviseur.

Méthode	Fonction	Vérification
La constructeur "User"	Permettre de créer les compte d' utilisateur	Vérifier que les données sont bien créées et enregistrées dans la base de données.
Une méthode "updateUser"	Permettre au superviseur de modifier les données des utilisateurs	Vérifier que les données des utilisateurs sont bien modifiées lorsque l'on appelle cette méthode
Une méthode de création de projet "creatProject"	permettre au superviseur de créer un projet.	Vérifier que les données des projets sont bien créées par cette méthode lorsqu'elle est appelée
Une méthode "updateProject ".	permettre au superviseur de modifier les attributs de l'utilisateur	Vérifier que les données des projets sont bien modifiées par cette méthode.
Une méthode "createMetier"	permettre au superviseur de créer des métiers.	Vérifier que les données des métiers sont bien créées par cette méthode
Une méthode "updateMetier"	permettre au superviseur de modifier les données des metiers.	Vérifier que les données des métiers sont bien modifiées dans la base des données par cette méthode
Une méthode "giveRole"	permettre au superviseur de affecter les projets au responsable de projet.	Vérifier que les données des responsables de projet et les données des projets sont bien associées.

- Responsable des projet possède les méthodes suivantes, vérifier les méthode ne peuvent être appelées que par le responsable des projets et superviseur

Méthode	Fonction	Vérification
Une méthode "createLot"	Permettre au responsable de projet de créer les lots	Vérifier que les données des lots sont bien créés
Une méthode "updateLot"	Permettre au responsable de projet de modifier les informations des lots	Vérifier que les données des lots sont bien modifiées.
Une méthode "giveLotToRespo"	Permettre au responsable de projet d'affecter les lots au responsable des lots	Vérifier que les données des lots et des responsables des lots sont associées
Une méthode "terminateLot"	Permettre au responsable de projet de clôturer les lots	Vérifier que les données des lots ne peuvent plus être modifiées après utilisation de cette méthode

- Responsable des lots possède les méthodes suivantes, vérifier les méthode ne peuvent être appelées que par le responsable des lots et superviseur

Méthode	Fonction	Vérification
Une méthode "createActivity"	Permettre de créer les activités	Vérifier que les données des activités sont bien créées
Une méthode "updateActivity"	Permettre de modifier les données des activités	Vérifier que les données des activités dans la base des données sont bien modifiées
Une méthode "giveActivity"	Permettre d'affecter les activité à l'utilisateur classique	Vérifier que les données des activités et des utilisateurs sont bien associées.
Une méthode "terminateActivity"	Permettre de clôturer les activités	Vérifier que les données des activités ne peuvent plus être modifiées après utilisation de cette

		méthode
--	--	---------

- Le responsable des métiers possède les méthodes suivantes. Vérifier que ces méthodes ne peuvent être appelées que par responsable des métier et superviseur.

Méthode	Fonction	Vérification
Une méthode "setInfoUser"	Permettre saisir les informations pour les utilisateurs	Vérification des données des utilisateurs sont bien créées et enregistrées sur la base de donnée

- Utilisateur classique possède les méthodes suivantes, vérifier que ces méthodes peuvent appelées par toutes les utilisateur

Méthode	Fonction	Vérification
Une méthode "setInfoWork"	Permettre de saisir les informations des travaux des utilisateurs	Vérification des données des informations des travaux sont bien créées et enregistrées sur la base de donnée
"getNom"	Permet d'obtenir les nom d' utilisateur	Vérifier bien que la sortir le nom de l'utilisateur
...

Les trois points représentent les méthodes de getInfo, ils ont la même façon de vérifications que la méthode "getNom".

Test de la partie Projet

- Vérifier que les attributs du Projet sont bien définis:
 - Nom de projet

- Nom du client
- Description du projet
- Budget accordé au projet
- Statut du projet : En cours/clos
- Ci : identifiant du projet de la forme PPPP
- Dates de début et fin de projet réelles ainsi que prévisionnelles
- Temps de projet (temps écoulé depuis le début du projet)
- Liste des managers du projet
- Liste des lots du projet

Attributs des lots :

- Ci : identifiant du lot de la forme LL (un nombre)
- Statut du lot: En cours/clos
- Dates de début et fin de lot prévisionnelles
- Durée actuelle du lot
- Budget réservé au lot
- Achats totaux pour le lot
- Liste des managers du lot
- Liste des activités du lot

Attributs des activités :

- Ci : identifiant de l'activité de la forme AAAA (un texte)
- Statut de l'activité : En cours/clos
- Description textuelle de l'activité (30 caractères minimum)
- Dates de début et fin d'activité réelles ainsi que prévisionnelles
- Budget prévu pour l'activité
- Total des achats effectués
- Liste des managers de l'activité
- Liste du personnel lié à l'activité

- Vérifier que les méthodes sont bien définies et qu'elles sont bien fonctionnelles:

Méthode	Fonction	Vérification
---------	----------	--------------

add_lot()	Créer un nouveau lot dans le projet	Vérifier que le lot a bien été créé
remove_lot(ci LL)	Retirer le lot LL du projet	Vérifier que le lot est bien retiré et qu'aucun autre n'est affecté
add_activité()	Créer une nouvelle activité dans le lot	Vérifier que l'activité a bien été créée
remove_activité(ci AAAA)	Retirer l'activité LL du lot	Vérifier que l'activité est bien retirée et qu'aucune autre n'est affectée
get_ci()	Retourne le ci du projet/lot/activité	Vérifier que c'est le bon ci qui est retourné
set_ci()	Modifie le ci du projet/lot/activité	Vérifier que le ci est unique
get_status()	Donne le statut du projet/lot/activité	Vérifier que c'est le bon statut qui est retourné
set_status(status)	Modifie le statut du projet/lot/activité	Vérifier la bonne modification du statut
get_list_lot()	Retourne la liste des lots d'un projet	Vérifier que tous les lots sont bien retournés
set_list_lot()	Met tous les lots dans une liste	Vérifier que les lots sont tous présents dans la liste
get_client()	Retourne le nom du client	Vérifier que c'est le bon nom
set_client()	Modifie le nom du client du projet	Vérifier la bonne modification du nom du client
set_description(description)	Modifie la description du projet/activité	Vérifier la bonne modification de la description. Pour la description de l'activité, elle doit faire 30 caractères minimum : le vérifier.
get_description()	Retourne la description du	Vérifier que c'est la bonne

	projet(activité)	description
get_exp_init_date() get_exp_end_date() get_real_init_date() get_real_end_date()	Retourne la date de début/fin de projet/lot/activité réelles/prévisionnelles	Vérifier que c'est les bonnes dates
set_exp_init_date() set_exp_end_date() set_real_init_date() set_real_end_date()	Modifie la date de début/fin de projet/lot/activité réelles/prévisionnelles	Vérifier que les modifications ont lieu. Ne pas avoir les dates d'activités plus grandes que celles du lot et les dates du lot plus grandes que celles du projet.
get_list_managers()	Retourne la liste des managers du projet/lot/activité	Vérifier qu'ils sont tous retournés.
set_list_managers()	Modifie la liste des managers du projet/lot/activité	Vérifier la bonne modification de la liste.
get_budget()	Retourne le budget réservé au projet/lot/activité	Vérifier que c'est la bonne valeur
set_budget()	Modifie la valeur du budget réservé au projet/lot/activité	Vérifier la bonne modification du budget. Il ne faut pas que le budget d'une activité soit supérieur à celui du lot. Il ne faut pas que le budget d'un lot soit supérieur à celui du projet

Test de base donnée

On a accès à la base de données soit en mode écriture, soit en mode lecture. On a donc des méthodes permettant d'interagir avec la database :

- **addUser** : ajoute un utilisateur dans la table des utilisateurs : elle prend en

paramètre un objet de type User.

- **addProjet** : ajoute un projet dans la table des projets : elle prend en paramètre un objet de type Project.
- **addLot** : ajoute un projet dans la table des lots : elle prend en paramètre un objet de type Lot.
- **addActivity** : ajoute une activité dans la table des activités : elle prend en paramètre un objet de type Activity.
- **update** : mise à jour d'une donnée quelconque. Elle prend en paramètre la table à modifier, les colonnes de la table à modifier et les nouvelles valeurs.

Pour les méthodes ,si l'objet en paramètre admet un champ vide ou non conforme, alors aucune action ne sera faite et une spécifique est renvoyée.

Pour la méthode **update**, la table à modifier et les colonnes à modifier doivent évidemment exister, sinon une erreur doit exister. Les nouvelles valeurs de mise à jour doivent être conformes (sur le type de données), sinon une erreur est renvoyée.