

Tests unitaires de méthodes de la classe Verif

verifIdEvenement :

Test unitaire de : *public CodeErreur verifIdEvenement(int unId);*

Résultat :

- null unId est conforme
- CodeErreur.ID_NEGATIF si unId est inférieur ou égal à 0

Paramètre unId	Résultat CodeErreur
1	null
3	null
0	ID_NEGATIF
-1	ID_NEGATIF
-3	ID_NEGATIF

verifNomEvenement :

Test unitaire de : `public CodeErreur verifNomEvenement (String unNom);`

Initialisation : `this.TAILLE_MAX_NOM_EVENEMENT = 20`

Résultat :

- null si unNom est conforme
- NOM_ESPACE_EN_TROP si unNom commence ou termine par un espace
- NOM_TROP_LONG si unNom dépasse la taille maximum
- NOM_NULL si unNom est null
- NOM_VIDE si unNom est une chaîne vide

Paramètre unNom	Résultat CodeErreur
"Nom de l'événement" (18 caractères)	null
" espaces avant" (15 caractères)	NOM_ESPACE_EN_TROP
"espaces après " (15 caractères)	NOM_ESPACE_EN_TROP
"Nom d'événement 19c" (19 caractères)	null
"Nom d'événement 20cc" (20 caractères)	null
"Nom d'événement 21ccc" (21 caractères)	NOM_TROP_LONG
null	NOM_NULL
" "	NOM_VIDE

verifDescriptifEvenement:

Test unitaire de : `public CodeErreur verifDescriptifEvenement(String unDescriptif);`

Initialisation : `this.TAILLE_MAX_DESCRIPTORIF_EVENEMENT = 20`

Résultat :

- null si unDescriptif est conforme
- DESCRIPTIF_ESPACE_EN_TROP si unDescriptif commence ou termine par un espace
- DESCRIPTIF_TROP_LONG si unDescriptif dépasse la taille maximum
- DESCRIPTIF_NULL si unDescriptif est null

Paramètre unDescriptif	Résultat CodeErreur
"Descriptif" (10 caractères)	null
" "	null
" espaces avant" (15 caractères)	DESCRIPTIF_ESPACE_EN_TROP
"espaces après " (15 caractères)	DESCRIPTIF_ESPACE_EN_TROP
"Descriptif dix-neuf" (19 caractères)	null
"Descriptif vingt car" (20 caractères)	null
"Descriptif vingt et 1" (21 caractères)	DESCRIPTIF_TROP_LONG
null	DESCRIPTIF_NULL

verifDateEvenement :

Test unitaire de : *public CodeErreur verifDateEvenement(Date uneDate);*

Résultat :

- null si uneDate est conforme
- DATE_NULL si uneDate est null
- DATE_PASSEE si uneDate est une date déjà passée

Paramètre unDescriptif	Résultat CodeErreur
/*25/11/22 19:30*/ uneDate = Date(2022-1900, 11, 25, 19, 30);	null
/*Date déjà passée 25/11/21 19:30*/ uneDate = Date(2021-1900, 11, 25, 19, 30);	DATE_PASSEE
null	DATE_NULL

verifLieuEvenement :

Test unitaire de : `public CodeErreur setLieu(String unLieu);`

Initialisation : `this.TAILLE_MAX_LIEU_EVENEMENT = 20`

Résultat :

- null si unLieu est conforme
- LIEU_ESPACE_EN_TROP si unLieu commence ou termine par un espace
- LIEU_TROP_LONG si unLieu dépasse la taille maximum
- LIEU_NULL si unLieu est null
- LIEU_VIDE si unLieu est une chaîne vide

Paramètre unLieu	Résultat CodeErreur
"5 rue Paix, Brest"	null
" espaces avant" (15 caractères)	LIEU_ESPACE_EN_TROP
"espaces après " (15 caractères)	LIEU_ESPACE_EN_TROP
"nom du lieu 19 cara" (19 caractères)	null
"un nom de lieu vingt" (20 caractères)	null
"nom du lieu vingtetun" (21 caractères)	LIEU_TROP_LONG
null	LIEU_NULL
""	LIEU_VIDE

verifNbMaxPersonnesEvenement :

Test unitaire de : *public CodeErreur verifNbMaxPersonnesEvenement (int unNbMaxPersonnes);*

Résultat :

- null si unNbMaxPersonnes est conforme
- NB_MAX_PERSONNES_TROP_PETIT si unNbMaxPersonnes est inférieur ou égal à 2

Paramètre unNbMaxPersonnes	Résultat CodeErreur
30	null
2	null
1	NB_MAX_PERSONNES_TROP_PETIT
0	NB_MAX_PERSONNES_TROP_PETIT
-2	NB_MAX_PERSONNES_TROP_PETIT

verifTypeEvenement :

Test unitaire de : *public CodeErreur verifTypeEvenement (TypeEvenement unType);*

Résultat :

- null si unType est conforme
- TYPE_NULL si unType est null

Paramètre unType	Résultat CodeErreur
TypeEvenement.REPAS	null
null	TYPE_NULL

verifPseudoMembre

Test unitaire de : `public CodeErreur verifPseudoMembre(String unPseudo)`

Initialisation : `this.TAILLE_MAX_PSEUDO_MEMBRE = 10`

Résultat :

- null si unPseudo est conforme
- PSEUDO_TROP_LONG si le pseudo est trop long
- CodeErreur.PSEUDO_NULL si le pseudo est null
- CodeErreur.PSEUDO_VIDE si le pseudo est vide
- CodeErreur.PSEUDO_INVALIDE si le pseudo contient : espace, <, >, |, :, ", *, ?, /, \

Paramètre unPseudo	Résultat CodeErreur
"Test"	null
"123"	null
"test-"	null
"testé"	null
null	PSEUDO_NULL
""	PSEUDO_VIDE
"un espace"	PSEUDO_INVALIDE
"pseudo-9c" (9 caractères)	null
"pseudo_10c" (10 caractères)	null
"pseudo-onze" (11 caractères)	PSEUDO_TROP_LONG
"<"	PSEUDO_INVALIDE
">"	PSEUDO_INVALIDE
" "	PSEUDO_INVALIDE
":"	PSEUDO_INVALIDE
"'" (guillemet)	PSEUDO_INVALIDE
"*"	PSEUDO_INVALIDE
"?"	PSEUDO_INVALIDE
"/"	PSEUDO_INVALIDE
"\"	PSEUDO_INVALIDE
" " (espace)	PSEUDO_INVALIDE

verifIdentiteMembre

Test unitaire de : `public CodeErreur verifIdentiteMembre(String designation, int numero)`

Initialisation :

- `this.TAILLE_MAX_NOM_MEMBRE=10`
- `this.TAILLE_MAX_PRENOM_MEMBRE=10`

Résultat :

- null si la designation est conforme
- si numero = 0
- `CodeErreur.NOM_TROP_GRAND` si le nom est trop grand
 - `CodeErreur.NOM_NULL` si le nom est null
 - `CodeErreur.NOM_VIDE` si le nom est vide
 - `CodeErreur.NOM_HORS_ALPHABET` si le nom contient des caractères autres que les lettres de l'alphabet
- si numero != 0
- `CodeErreur.PRENOM_TROP_GRAND` si le nom est trop grand
 - `CodeErreur.PRENOM_NULL` si le nom est null
 - `CodeErreur.PRENOM_VIDE` si le nom est vide
 - `CodeErreur.PRENOM_HORS_ALPHABET` si le nom contient des caractères autres que les lettres de l'alphabet

Paramètres (designation, numero)	Résultat CodeErreur
("Pierre", 0)	null
("Nom", 0)	null
("NomNeufCa",0) (9 caractères)	null
("NomDixCara",0) (10 caractères)	null
("NomOnzeCara",0) (11 caractères)	NOM_TROP_GRAND
(null, 0)	NOM_NULL
("", 0)	NOM_VIDE
("un espace", 0)	NOM_HORS_ALPHABET
("Accént", 0)	NOM_HORS_ALPHABET
("Chiffre1", 0)	NOM_HORS_ALPHABET
("Pre-nom", 0)	NOM_HORS_ALPHABET
("Pre*nom",0)	NOM_HORS_ALPHABET

("Pierre", 1)	null
("PreNom", 1)	null
("PrenomNeu", 1) (9 caractères)	null
("PrenomDixC", 1) (10 caractères)	null
("PrenomOnzeC", 1) (11 caractères)	PRENOM_TROP_GRAND
(null, 1)	PRENOM_NULL
("", 1)	PRENOM_VIDE
("un espace", 1)	PRENOM_HORS_ALPHABET
("Accént", 1)	PRENOM_HORS_ALPHABET
("Chiffre1", 1)	PRENOM_HORS_ALPHABET
("Pre-nom", 1)	PRENOM_HORS_ALPHABET
("Pre*nom", 1)	PRENOM_HORS_ALPHABET

verifLieuxMembre

Test unitaire de : `public CodeErreur verifLieuxMembre(String lieu, int numero)`

Initialisation : `this.TAILLE_MAX_LIEU_MEMBRE=10`

Résultat :

- null si le lieu est conforme
- si numero = 0
 - CodeErreur.LIEU_NULL si le lieu de naissance est null
 - CodeErreur.LIEU_NOM_INCORRECT si le nom du lieu de naissance comporte un caractère spécial
 - CodeErreur.LIEU_TAILLE_INCORRECTE si le lieu de naissance ne comprend pas le nombre de caractères requis
- si numero != 0
 - CodeErreur.VILLE_NOM_NULL si la ville est null
 - CodeErreur.VILLE_NOM_INCORRECT si le nom de la ville comporte un caractère spécial
 - CodeErreur.VILLE_TAILLE_INCORRECTE si la ville ne comprend pas le nombre de caractères requis

Paramètre (lieu, numero)	Résultat CodeErreur
("Lieu", 0)	null
("LieuNeufC", 0) (9 caractères)	null
("LieuDixCar", 0) (10 caractères)	null
("LieuOnzeCar", 0) (11 caractères)	LIEU_TAILLE_INCORRECTE
("Accént", 0)	LIEU_NOM_INCORRECT
(null, 0)	LIEU_NULL
("un espace", 0)	null
("chiffre1", 0)	LIEU_NOM_INCORRECT
("", 0)	LIEU_TAILLE_INCORRECTE
("un,lieu",0)	LIEU_NOM_INCORRECT
("Ville", 1)	null
("VilleNeuf", 1) (9 caractères)	null
("VilleDixCa", 1) (10 caractères)	null

("VilleOnzeCa", 1) (<i>11 caractères</i>)	VILLE_TAILLE_INCORRECTE
("Accént", 1)	VILLE_NOM_INCORRECT
(null, 1)	VILLE_NULL
("un espace", 1)	null
("chiffre1", 1)	VILLE_NOM_INCORRECT
("", 1)	VILLE_TAILLE_INCORRECTE
("une,ville",1)	VILLE_NOM_INCORRECT

verifDateMembre

Test unitaire de : `public CodeErreur verifDateMembre(String uneDateNaissance)`

Résultat :

- null si la date de naissance est conforme
- Code_Erreur.DATE_NULL si la date est null
- CodeErreur.DATE_IMPOSSIBLE si la date de naissance est plus grande que la date courante
- CodeErreur.DATE_AGE_IMPOSSIBLE si l'âge du membre est supérieur à 130 ans

Paramètre uneDateNaissance	Résultat CodeErreur
<code>/*01/01/2001 01:01*/ uneDate = Date(2001-1900, 1, 1, 1, 1);</code>	null
<code>/*Date plus grande que la date actuelle 01/01/2050 01:01 */ uneDate = Date(2023-1900, 1, 1, 1, 1);</code>	DATE_IMPOSSIBLE
<code>/*Age > 130 ans 01/01/1890 01:01 */ uneDate = Date(1890-1900, 1, 1, 1, 1);</code>	DATE_AGE_IMPOSSIBLE
null	DATE_NULL

verifMailMembre

Test unitaire de : `public CodeErreur verifMailMembre(String unMail)`

Initialisation : `this.TAILLE_MAX_MAIL_MEMBRE=20`

Résultat :

- null si la définition du mail est un succès
- CodeErreur.MAIL_NULL le mail est null
- CodeErreur.MAIL_VIDE si le mail est vide
- CodeErreur.MAIL_TROP_LONG si le mail est trop long
- CodeErreur.MAIL_INVALIDE si le mail ne correspond pas à une adresse mail

Paramètre unMail	Résultat CodeErreur
"tof.bo@example.com"	null
"tof@example.me.org"	null
"tof@example.com"	null
null	MAIL_NULL
"	MAIL_VIDE
"to19@example.me.org" (19 caractères)	null
"tof20@example.me.org" (20 caractères)	null
"tof.21@example.me.org" (21 caractères)	MAIL_TROP_LONG
"tof@.com"	MAIL_INVALIDE
"tof@exemple"	MAIL_INVALIDE
"tof.example.com"	MAIL_INVALIDE
"tof..bob@example.com"	MAIL_INVALIDE
"@example.me.org"	MAIL_INVALIDE
"tof#@exemple.me.org"	MAIL_INVALIDE

setMotDePasse

Test unitaire de : `public CodeErreur setMotDePasse(String unMotDePasse)`

Initialisation : `this.TAILLE_MAX_MDP_MEMBRE = 10;`

Résultat :

- null si le mot de passe est conforme
- CodeErreur.MDP_NULL si le mot de passe est null
- CodeErreur.MDP_VIDE si le mot de passe n'est composé d'aucun caractère
- CodeErreur.MDP_TAILLE_INCORRECTE si le mot de passe est trop court ou trop long
- CodeErreur.MDP_INCORRECT si le mot de passe n'est pas assez sécurisant (manque majuscule, minuscule, chiffre et/ou caractère spécial)

Paramètre unMotDePasse	Résultat CodeErreur
"Mdp1!a"	null
"mdp1!a"	MDP_INCORRECT (pas de majuscule)
"Mdp!aa"	MDP_INCORRECT (pas de chiffre)
"MDP1!A"	MDP_INCORRECT (pas de minuscule)
"Mdp1aa"	MDP_INCORRECT (pas de caractère spécial)
"mdpaaa"	MDP_INCORRECT (que des minuscules)
null	MDP_NULL
""	MDP_TAILLE_INCORRECTE
"Mdp5!" (5 caractères)	MDP_TAILLE_INCORRECTE
"Mdp6c!" (6 caractères)	null
"Mdp9cara!" (9 caractères)	null
"Mdp10cara!" (10 caractères)	null
"Mdp11cara!!" (11 caractères)	MDP_TAILLE_INCORRECTE

Tests unitaires de méthodes de la classe BaseDeDonnees

updateMembres :

Test unitaire de : `public void updateMembres(Gestion g);`

Initialisation :

`g.setListeMembres(new ArrayList<Membre>());`

Résultat :

Procédure	MEMBRE de la base de données	ListeMembres de Gestion
Aucun problème avec MEMBRE de la BDD vide	[]	[]
Aucun problème avec MEMBRE de la BDD remplie	(“p1”, “n”, “pr”, “v”, “2001-09-07”, “v”, “m@m.m”, “mdppppppP5”) (“p2”, “nom”, “prenom”, “lieu”, “2021-08-08”, “ville”, “mail@mail.m ail”, “motDePasse7”) (“p3”, “nom”, “prenom”, “lieu”, “2021-01-10”, “ville”, “ ma@ma.ma ”, “motDePasse7!”)	(“p1”, “n”, “pr”, “v”, “2001-09-07”, “v”, “m@m.m”, “mdppppppP5”) (“p2”, “nom”, “prenom”, “lieu”, “2021-08-08”, “ville”, “m ail@mail.mail”, “motDePa sse7”) (“p3”, “nom”, “prenom”, “lieu”, “2021-01-10”, “ville”, “ ma @ma.ma ”, “motDePasse 7!”)
Perte de connexion ou problème d'exécution de la requête dans la BDD	(“p1”, “n”, “pr”, “v”, “2001-09-07”, “v”, “m@m.m”, “mdppppppP5”) (“p2”, “nom”, “prenom”, “lieu”, “2021-08-08”, “ville”, “mail@mail.m ail”, “motDePasse7”) (“p3”, “nom”, “prenom”, “lieu”, “2021-01-10”, “ville”, “ ma@ma.ma ”, “motDePasse7!”)	[]

updateEvenements :

Test unitaire de : `public void updateEvenements(Gestion g);`

Initialisation :

`g.setListeEvenements(new ArrayList<Evenement>());`

Résultat :

Procédure	EVENEMENT de la base de données	ListeEvenements de Gestion
Aucun problème avec EVENEMENT de la BDD vide	[]	[]
Aucun problème avec EVENEMENT de la BDD remplie	("1","origodi",",",",",",",2024-01-14", "lieu","2","ANIMATION") ("2","evenement2",",",",",",2023-05-19", "lieu","54","AG") ("3","evenement3",",",",",",2024-04-15", "lieu","33","AG")	("1","origodi",",",",",",2024-01-14", "lieu","2","ANIMATION") ("2","evenement2",",",",",",2023-05-19", "lieu","54","AG") ("3","evenement3",",",",",",2024-04-15", "lieu","33","AG")
Perte de connexion ou problème d'exécution de la requête dans la BDD	("1","origodi",",",",",",2024-01-14", "lieu","2","ANIMATION") ("2","evenement2",",",",",",2023-05-19", "lieu","54","AG") ("3","evenement3",",",",",",2024-04-15", "lieu","33","AG")	[]

updateParticipation :

Test unitaire de : `public void updateParticipation(Gestion g);`

Initialisation :

`g.setListeParticipations(new ArrayList<Participation>());`

Résultat :

Procédure	PARTICIPATION de la base de données	ListeParticipations de Gestion
Aucun problème avec PARTICIPATION de la BDD vide	[]	[]
Aucun problème avec PARTICIPATION de la BDD remplie	("3", "p1", "5", "") ("3", "p2", "7", "") ("2", "p3", "15", "infos")	("3", "p1", "5", "") ("3", "p2", "7", "") ("2", "p3", "15", "infos")
Perte de connexion ou problème d'exécution de la requête dans la BDD	("3", "p1", "5", "") ("3", "p2", "7", "") ("2", "p3", "15", "infos")	[]

membrePresent :

Test unitaire de : *public boolean membrePresent(String unPseudo) throws SQLException;*

Initialisation :

Éléments de la table MEMBRE de la base de données :

- ("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*")
- ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*")
- ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")

Résultat :

Paramètre unPseudo	Résultat boolean
"p1"	true
"pseudonexistent"	false

evenementPresent :

Test unitaire de : *public boolean evenementPresent(int unld) throws SQLException;*

Initialisation :

Élément de la table EVENEMENT de la base de données :

- ("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION")
- ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG")
- ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")

Résultat :

Paramètre unld	Résultat boolean
1	true
71	false

participationPresente :

Test unitaire de : *public boolean participationPresente(int id, String pseudo) throws SQLException;*

Initialisation :

Élément de la table PARTICIPATION de la base de données :

- ("2", "p1", "14", "")
- ("3", "p2", "5", "")
- ("1", "p1", "2", "")

Résultat :

Paramètre id, pseudo	Résultat boolean
2, p1	true
3, p1	false

ajouterMembre :

Test unitaire de : *public boolean ajouterMembre(Gestion g, Membre m)*

Initialisation :

Élément de la table MEMBRE de données :

- ("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*")
- ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*")
- ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")

Résultat :

Procédure	Paramètre m	Modification MEMBRE de la BDD	Résultat boolean
Aucun problème	("p4", "nom", "prenom", "lieu", "2021-07-21", "Brest", "ma@ma.ma", "motDePasse7!")	("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*") ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*") ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!") ("p4", "nom", "prenom", "lieu", "2021-07-21", "Brest", "ma@ma.ma", "motDePasse7!")	true
Perte de connexion ou problème d'exécution de la requête dans la BDD	("p4", "nom", "prenom", "lieu", "2021-07-21", "Brest", "ma@ma.ma", "motDePasse7!")	("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*") ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*") ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")	false

ajouterEvenement :

Test unitaire de : *public boolean ajouterEvenement(Gestion g, Evenement e)*

Initialisation :

Élément de la table EVENEMENT de la base de données :

- ("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION")
- ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG")
- ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")

Résultat :

Procédure	Paramètre e	Modification EVENEMENT de la BDD	Résultat boolean
Aucun problème	("4", "evenement4", "", "", "2026-04-02", "lieu", "6", "AG")	("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG") ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG") ("4", "evenement4", "", "", "2026-04-02", "lieu", "6", "AG")	true
Perte de connexion ou problème d'exécution de la requête dans la BDD	("4", "evenement4", "", "", "2026-04-02", "lieu", "6", "AG")	("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG") ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")	false

ajouterParticipation :

Test unitaire de : *public boolean ajouterParticipation(Gestion g, Participation p)*

Initialisation :

Élément de la table PARTICIPATION de la base de données :

- ("2", "p1", "14", "")
- ("3", "p2", "5", "")
- ("1", "p1", "2", "")

Résultat :

Procédure	Paramètre p	Modification PARTICIPATION de la BDD	Résultat boolean
Aucun problème	("3", "p1", "17", "")	("2", "p1", "14", "") ("3", "p2", "5", "") ("1", "p1", "2", "") ("3", "p1", "17", "")	true
Perte de connexion ou problème d'exécution de la requête dans la BDD	("3", "p1", "17", "")	("2", "p1", "14", "") ("3", "p2", "5", "") ("1", "p1", "2", "")	false

supprimerEvenement :

Test unitaire de : *public boolean supprimerEvenement(Gestion g, int unld)*

Initialisation :

Élément de la table EVENEMENT de la base de données :

- ("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION")
- ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG")
- ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")

Élément de la table PARTICIPATION de la base de données :

- ("3", "p1", "5", "")
- ("3", "p2", "7", "")
- ("2", "p3", "15", "infos")

Résultat :

Procédure	Paramètre unld	Modification EVENEMENT de la BDD	Modification PARTICIPATION de la BDD	Résultat boolean
Aucun problème	3	("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG")	("2", "p3", "15", "infos")	true
unld n'est pas présent dans la table	71	("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG") ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")	("3", "p1", "5", "") ("3", "p2", "7", "") ("2", "p3", "15", "infos")	false
Perte de connexion ou problème d'exécution de la requête dans la BDD	1	("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG") ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")	("3", "p1", "5", "") ("3", "p2", "7", "") ("2", "p3", "15", "infos")	false

supprimerMembre :

Test unitaire de : *public boolean supprimerMembre(Gestion g, String unPseudo)*

Initialisation :

Élément de la table MEMBRE de la base de données :

- ("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*")
- ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePss7*")
- ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")

Élément de la table PARTICIPATION de la base de données :

- ("3", "p1", "5", "")
- ("3", "p2", "7", "")
- ("2", "p3", "15", "infos")

Résultat :

Procédure	Paramètre unPseudo	Modification MEMBRE de la BDD	Modification PARTICIPATION de la BDD	Résultat boolean
Aucun problème	"p1"	("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePss7*"), ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")	("3", "p2", "7", "") ("2", "p3", "15", "infos")	true
unPseudo n'est pas présent dans la table	"pseudoInexistant"	("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*") ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePss7*") ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")	("3", "p1", "5", "") ("3", "p2", "7", "") ("2", "p3", "15", "infos")	false
Perte de connexion ou problème d'exécution de la requête dans la BDD	"p1"	("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*") ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePss7*") ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")	("3", "p1", "5", "") ("3", "p2", "7", "") ("2", "p3", "15", "infos")	false

supprimerParticipation :

Test unitaire de : *public boolean supprimerParticipation(Gestion g, Participation p)*

Initialisation :

Élément de la table PARTICIPATION de la base de données :

- ("2", "p1", "14", "")
- ("3", "p2", "5", "")
- ("1", "p1", "2", "")

Résultat :

Procédure	Paramètre p	Modification PARTICIPATION de la BDD	Résultat boolean
Aucun problème	("2", "p1", "14", "")	("3", "p2", "5", "") ("1", "p1", "2", "")	true
Perte de connexion ou problème d'exécution de la requête dans la BDD	("2", "p1", "14", "")	("2", "p1", "14", "") ("3", "p2", "5", "") ("1", "p1", "2", "")	false

modifierMembre :

Test unitaire de : *public boolean modifierMembre(Gestion g, Membre m)*

Initialisation :

Élément de la table MEMBRE de la base de données :

- ("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*")
- ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*")
- ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")

Résultat :

Procédure	Paramètre m	Modification MEMBRE de la BDD	Résultat boolean
Aucun problème	("p2", "nouveau", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*")	("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*") ("p2", "nouveau", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*") ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")	true
unPseudo n'est pas présent dans la table	("pseudoInexistant", "nouveau", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*")	("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*") ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*") ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")	false
Perte de connexion ou problème d'exécution de la requête dans la BDD	("p2", "nouveau", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*")	("p1", "n", "pr", "v", "2001-09-07", "v", "m@m.m", "mdppppppP5*") ("p2", "nom", "prenom", "lieu", "2021-08-08", "ville", "mail@mail.mail", "motDePasse7*") ("p3", "nom", "prenom", "lieu", "2021-01-10", "ville", "ma@ma.ma", "motDePasse7!")	false

modifierParticipation :

Test unitaire de : *public boolean modifierParticipation(Gestion g, Participation p)*

Initialisation :

Élément de la table PARTICIPATION de la base de données :

- ("2", "p1", "14", "")
- ("3", "p2", "5", "")
- ("1", "p1", "2", "")

Résultat :

Procédure	Paramètre p	Modification PARTICIPATION de la BDD	Résultat boolean
Aucun problème	("2", "p1", "10", "")	("2", "p1", "10", "") ("3", "p2", "5", "") ("1", "p1", "2", "")	true
Perte de connexion ou problème d'exécution de la requête dans la BDD	("2", "p1", "10", "")	("2", "p1", "14", "") ("3", "p2", "5", "") ("1", "p1", "2", "")	false

modifierEvenement :

Test unitaire de : *public boolean modifierEvenement(Gestion g, Evenement e)*

Initialisation :

Élément de la table EVENEMENT de la base de la base de données :

- ("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION")
- ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG")
- ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")

Résultat :

Procédure	Paramètre e	Modification EVENEMENT de la BDD	Résultat boolean
Aucun problème	("1", "nouveauNom", "", "", "2024-01-14", "Brest", "2", "ANIMATION")	("1", "nouveauNom", "", "", "2024-01-14", "Brest", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG") ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")	true
unPseudo n'est pas présent dans la table	("71", "nouveauNom", "", "", "2024-01-14", "Brest", "2", "ANIMATION")	("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG") ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")	false
Perte de connexion ou problème d'exécution de la requête dans la BDD	("1", "nouveauNom", "", "", "2024-01-14", "Brest", "2", "ANIMATION")	("1", "origodi", "", "", "2024-01-14", "lieu", "2", "ANIMATION") ("2", "evenement2", "", "", "2023-05-19", "lieu", "54", "AG") ("3", "evenement3", "", "", "2024-04-15", "lieu", "33", "AG")	false

Tests d'intégration

Manipulation de membres, d'évènements et de participations à la chaîne

Initialisation: Listes des membres, des évènements et des participations vides

Participations:

(ajouts, modifications et suppressions des participations peut être fait avec l'interface web)

Évènement 1: Membres (Alpha, Bravo, Charlie)

Évènement 2: Membres (Alpha, Bravo)

Évènement 3: Membres (Charlie)

Appel	Retour
getListeMembres()	{}
ajouterMembre("Alpha", "ANDRE", "Tom", "BREST", "28/01/1999", "BREST", "tomandre@gmail.com", "Tom29\$\$")	NO_ERROR
getListeMembres()	{"Alpha"}
ajouterMembre("Alpha", "ANDRE", "Tom", "", "28/01/2070", "BREST", "tomandre@gmail.com", "Tom29\$\$")	PSEUDO_DEJA_EXISTANT, LIEU_NAISSANCE_NULL, DATE_IMPOSSIBLE
getListeMembres()	{"Alpha"}
getListeEvenements()	{}
ajouterEvenement("1", "Assemblée Générale", "Election du bureau", "image.png", "15/08/2022", "BREST", "20", "AG")	NO_ERROR
getListeEvenements()	{"1"}
supprimerMembre("Bravo")	MEMBRE_INTROUVABLE
getListeMembres()	{"Alpha"}
ajouterMembre("Bravo", "GILLES", "Jules", "RENNES", "29/08/1994", "RENNES", "julesgilles@gmail.com", "Jules56\$\$")	NO_ERROR
getListeMembres()	{"Alpha", "Bravo"}
modifierMembre("Bravo", "GILLES", "Tom", "MARSEILLE", "28/01/2001", "MARSEILLE",)	NO_ERROR

"tomgilles@gmail.com", "Tom29\$\$")	
modifierMembre("Michel", "Angé", "Tom", "BREST", "28/01/2001", "BREST", "tomgilles@gmail.com", "Tom29\$\$")	MEMBRE_INTROUVABLE
getMembre("Alpha")	("Alpha", "ANDRE", "Tom", "BREST", "28/01/1999", "BREST", "tomandre@gmail.com", "Tom29\$\$")
getMembre("Bravo")	{("Bravo", "GILLES", "Tom", "MARSEILLE", "28/01/2001", "MARSEILLE", "tomgilles@gmail.com", "Tom29\$\$")}
supprimerMembre("Bravo")	NO_ERROR
getListeMembres()	{"Alpha"}
ajouterParticipation({"1",...}, {"Alpha",...}, 1, "")	NO_ERROR
getListeParticipations()	{"1", "Alpha"}
ajouterMembre("Bravo", "RENÉ", "Jean", "PARIS", "29/08/1994", "PARIS", "jeanrene@gmail.com", "Jean75\$\$")	NO_ERROR
getListeMembres()	{"Alpha", "Bravo"}
ajouterMembre("Charlie", "DUPUY", "Jack", "LYON", "18/02/1974", "LYON", "jackdupuy@orange.fr", "Jack69\$\$")	NO_ERROR
getListeMembres()	{"Alpha", "Bravo", "Charlie"}
getListeEvenements()	{"1"}
ajouterEvenement("2", "Point Chantier", "modifications des plannings", "planning.png", "17/08/2022", "RENNES", "10", "CHANTIER")	NO_ERROR
getListeEvenements()	{"1", "2"}
ajouterEvenement("3", "Soirée", "Soirée jeux de société", "image.png", "05/10/2023", "BREST", "30", "ANIMATION")	NO_ERROR
getListeEvenements()	{"1", "2", "3"}
ajouterParticipation({"1",...}, {"Bravo",...}, 1, "")	NO_ERROR
ajouterParticipation({"1",...}, {"Charlie",...}, 25, "")	NBINSCRIT_TROP_GRAND
ajouterParticipation({"1",...}, {"Charlie",...}, 4, "")	NO_ERROR
getListeParticipations()	{"1", "Alpha", "1", "Bravo", "1", "Charlie"}
supprimerParticipation({"2",...}, {"Bravo",...}, 1, "")	PARTICIPATION_INEXISTANTE

ajouterParticipation(null,{“Alpha”,...},1,“”)	EVENEMENT_NULL
ajouterParticipation({“2”,...},{“Alpha”,...},2,“”)	NO_ERROR
modifierParticipation({“3”,...},{“Alpha”,...},2,“”)	PARTICIPATION_INEXISTANTE
ajouterParticipation({“2”,...},{“Bravo”,...},1,“”)	NO_ERROR
ajouterParticipation({“2”,...},{“Michel”,...},2,“”)	MEMBRE_INEXISTANT
ajouterParticipation({“3”,...},{“Charlie”,...},3,“”)	NO_ERROR
getListeParticipations()	{“1”, “Alpha”, “1”, “Bravo”, “1”, “Charlie”, “2”, “Alpha”, “2”, “Bravo”, “3”, “Charlie”}
supprimerParticipation({“1”,...},{“Charlie”,...},4,“”)	NO_ERROR
getListeParticipations()	{“1”, “Alpha”, “1”, “Bravo”, “2”, “Alpha”, “2”, “Bravo”, “3”, “Charlie”}
supprimerEvenement(“2”)	NO_ERROR
getListeEvenements()	{“1”, “3”}
modifierParticipation({“2”,...},{“Charlie”,...},4,“”)	EVENEMENT_INTROUVABLE
modifierParticipation({“1”,...},{“Alpha”,...},4,“”)	NO_ERROR
getListeParticipations()	{“1”, “Alpha”, “1”, “Bravo”, “3”, “Charlie”} (La suppression précédente entraîne une modification des participations)
supprimerMembre(“Charlie”)	NO_ERROR
getListeMembres()	{“Alpha”, “Bravo”}
getListeParticipations()	{“1”, “Alpha”, “1”, “Bravo”} (Le membre Charlie est retiré, il n’y a désormais plus de participants à l’évènement 3)
modifierEvenement(“1”, “Soirée”, “Soirée jeux de société”, “image.png”, “05/10/2023”, “BREST”, “30”, “ANIMATION”)	NO_ERROR
getEvenement(“1”)	(“1”, “Soirée”, “Soirée jeux de société”, “image.png”, “05/10/2023”, “BREST”, “30”, “ANIMATION”)
getListeEvenements()	{“1”, “3”}

getListeParticipations()	{ "1", "Alpha", "1", "Bravo" (La modification d'un événement n'entraîne pas de changement dans les participations)
supprimerMembre("Bravo")	NO_ERROR
getListeMembres()	{ "Alpha" }
getListeParticipations()	{ "1", "Alpha" (Le membre Bravo est retiré, il n'y a désormais plus qu'un seul participant à l'évènement 1)
supprimerEvenement("3")	NO_ERROR
getListeEvenements()	{ "1" }
getListeParticipations()	{ "1", "Alpha" }
supprimerEvenement("1")	NO_ERROR
getListeEvenements()	{ }
getListeParticipations()	{ } (car il n'y a plus d'événements)
getListeMembres()	{ "Alpha" }
supprimerMembre("Alpha")	NO_ERROR
getListeMembres()	{ }
getListeEvenements()	{ }