## Fiche de Tests n° 23 – Type : FON

## *[prendre un trigramme pour le type de tests FON23 => Tests fonctionnels 23]*

**Type** : 🞏 fonctionnel 🞏 IU 🞏 données et BDD 🞏 Profilage

🞏 Charge 🞏 stress 🞏 volumétrie 🞏 sécurité contrôle

🞏 basculement récupération 🞏 configuration 🞏 tests d'installation

| **Descriptif** | **Description** |
| --- | --- |
| Version du projet | *xx* |
| Version du logicielle à tester | *xx* |
| Version des logiciels Tiers | *xx* |
| Nom du Testeur | *xx* |
| Date | *xx* |

**Description de la fonctionnalité** *[a tester…]***:**

*À compléter*

**Pré-requis:**

*[les données à avoir en début de tests : contenu des fichiers de configuration, contenu des tables en base de données, les paramètres pour un test sur une seule méthode...…]***:**

**Procédure/ déroulement :**

*[expliquer la procédure exacte de ce que vous devez faire pour tester….exemple pour une IHM, cliquez ici, cliquez la….entrez telles valeurs]*

**Résultat attendu :**

*[Expliquer le résultat attendu en cas de réussite]*

**Résultat obtenu:**

*[Expliquer le résultat observé]*

**Synthèse : 🞏 OK 🞏 KO**

**Explications :**

*[des commentaires sur les causes possibles si idée]*

| **Document  (version / date)** | **Disponibilité** | **Revu** | **Ressource** | **Notes** |
| --- | --- | --- | --- | --- |
| *Spécification d’exigences logicielles* | 🞏 Non | 🞏 Non |  |  |
| *Spécifications des cas d’utilisation* | 🞏 Non | 🞏 Non |  |  |
| *Réalisation des cas d'utilisation* | 🞏 Non | 🞏 Non |  |  |
| *Prototype d’interface utilisateur* | 🞏 Non | 🞏 Non |  |  |
| *Architecture logicielle* | 🞏 Non | 🞏 Non |  |  |
| *Plan de développement logiciel* | 🞏 Non | 🞏 Non |  |  |

# Exigences à tester

La liste suivante identifie les items, cas d’utilisation, exigences fonctionnelles et exigences non-fonctionnelles, qui ont été désignés comme cibles de test. Cette liste représente ce qui sera testé.

[Énumérer les principales exigences à tester]

# Stratégie de tests

[La stratégie de tests présente l’approche recommandée d’évaluation des cibles des tests. La précédente section, Exigences à tester, décrit de qui sera testé et cette section comment cela sera testé.

Fournir, pour chaque type de test, une description du test et comment il sera implémenté et exécuté.

Lorsque les tests ne sont pas implémentés et exécutés, expliquer en une phrase pourquoi.

La stratégie de tests consiste d’abord à identifier les techniques de tests et à identifier les critères de complétion des tests.

Les tests doivent être exécutés dans des environnements sécurisés avec des données connues et contrôlées.]

## Types de tests

### Tests fonctionnels

[Les tests fonctionnels portent sur les exigences fonctionnelles qui peuvent être retracées dans les cas d’utilisation ou les fonctions d’affaire et dans les règles d’affaire. Le but de ces tests est de vérifier la validité des données, de leur traitement et de leur récupération ainsi que des règles d’affaire. Ce type d’évaluation repose sur la technique de la boîte noire, c’est-à-dire, une vérification de l’application et de ses processus internes en l’utilisant avec l’interface utilisateur et en en analysant les résultats. Voici un exemple de stratégie proposée :]

|  |  |
| --- | --- |
| **Stratégie pour les tests fonctionnels** | |
| Objectif de test: | Vérifier que la fonctionnalité rencontre la bonne cible de test, incluant la navigation, la saisie de données, le traitement et la récupération. |
| Technique: | Exécuter les cas d’utilisation, chaque chemin du cas ou fonction en utilisant des données valides et non valides tout en vérifiant:   * Les résultats attendus avec des données valides. * Les messages d’erreur et d’avertissement lorsque des données non valides sont utilisées. * Que chaque règle d’affaire est appliquée correctement. |
| Critère de complétion: | * Tous les tests prévus ont été exécutés. * Toutes le anomalies identifiées ont été enregistrées. |
| Considérations: particulières | [Décrire les items ou soulever les questions internes ou externes qui ont un impact sur l’implémentation et l’exécution d’un test fonctionnel] |

### Tests d’interface utilisateur

[Les tests d’interfaces utilisateur évaluent l’interaction de l’utilisateur avec le logiciel. Le but de ces tests est de vérifier que l’interface utilisateur donne l’accès approprié aux fonctions des cibles de test. De plus, ces tests permettent de vérifier que les objets de l’interface utilisateur se comporte tel qu’attendu et qu’ils sont conformes aux normes du client ou de l’industrie.]

|  |  |
| --- | --- |
| **Stratégie pour les tests d’interface utilisateur** | |
| Objectif de test: | Évaluer:   * La navigation des cibles de test représentent adéquatement les fonctions d’affaire et les exigences, incluant d’écran à écran, de champ à champ, et l’utilisation des modes de déplacement tells les clés de tabulation, le mouvement de la souris et les clés d’accès rapide. * Les composants de l’écran et les caractéristiques tels les menus, la taille, la position, la cible de saisie et l’état sont conformes aux normes. |
| Technique: | Produire des tests pour chaque écran afin de vérifier la navigation et les états de chacun des écrans et composants de l’application. |
| Critère de complétion: | La vérification de chaque écran est concluante en se conformant à la référence ou à une norme acceptable. |
| Considérations: particulières | Il est possible que toutes les propriétés personnalisables ou des composants d’un tiers ne soient testées. |

### Tests de données et d’intégrité de base de données

[Les bases de données et les processus de base de données doivent être testés comme sous-systèmes à l’intérieur du projet. Ces sous-systèmes doivent être testés sans utiliser l’interface utilisateur comme interface des données. L’identification des outils et techniques pour effectuer les tests repose sur la nature du système de gestion de base de données (SGBD).]

|  |  |
| --- | --- |
| **Stratégie pour les tests de données et d’intégrité de base de données** | |
| Objectif de test: | Vérifier les méthodes d’accès à la base de données ainsi que les processus fonctionnent correctement et sans corruption de données. |
| Technique: | Appeler chaque méthode et processus d’accès à la base de données en alimentant chacun de données valides, de données invalides et de requêtes de données.  Inspecter la base de données pour vérifier que les données ont été chargées comme prévu, que toutes les opérations de la base de données s’exécutent normalement et réviser les données produites afin de s’assurer que les bonnes données ont été extraites et pour les bonnes raisons. |
| Critère de complétion: | Tous les processus et méthodes d’accès fonctionnent tels qu’ils ont été conçus sans corruption de données. |
| Considérations: particulières | Les tests peuvent nécessiter un environnement de développement du SGBD ou des pilotes pour charger ou modifier directement des données dans la base de données.  Les processus doivent être actives manuellement.  Des bases de données petites ou de taille minimale devraient être utilisées pour accroître la visibilité d’événements non acceptables. |

### Profilage de performance

[Le profilage de la performance est un test de performance où les temps de réponses, les taux de transaction et les autres exigences temporelles sont mesurées et évaluées. Le but du profilage de performance est de vérifier si les exigences de performances ont été atteintes. Il est implémenté et exécuté profiler et ajuster la performance des cibles de test en fonction de conditions comme la charge de travail et les configurations matérielles.

NOTE : Les transactions sont, ci-dessous, des transaction d’affaire logiques. « Ces transactions sont définies comme des cas d’utilisation qu’un acteur du système est présumé exécuter en utilisant une cible de test comme, par exemple, ajouter ou modifier un contrat.

|  |  |
| --- | --- |
| **Stratégie pour le profilage de performance** | |
| Objectif de test: | Vérifier la performance d’exécution de transactions désignées ou de fonctions d’affaires dans les conditions suivantes :  - Charge de travail normale  - Charge de travail dans le pire des cas |
| Technique: | Les procédures de test utilisées sont développées pour une évaluation cyclique de fonctions.  Modifier les fichiers de données pour accroître le nombre de transaction ou les scripts pour accroître le nombre d’itérations des transactions.  Les scripts doivent être exécutés sur une machine, le meilleur des cas pour référencer un seul utilisateur et une seule transaction et être répété avec plusieurs clients, virtuel ou réel selon les considérations particulières ci-dessous. |
| Critère de complétion: | Un seul utilisateur et une seule transaction : Complétion réussie des scripts de test sans problèmes et dans le temps requis prévu par transaction.  Plusieurs transactions et plusieurs utilisateurs: Complétion réussie des scripts de test sans problèmes et dans le temps requis prévu par transaction. |
| Considérations: particulières | Une évaluation de performance valable suppose une charge de travail en arrière pan sur le serveur :  Pour y arriver, différentes méthodes peuvent être utilisées :   * Émuler des transactions directement sur le serveur à partir de requêtes SQL. * Créer une charge utilisateur virtuelle, pour simuler plusieurs centaines de clients. Des outils d’émulation de terminal à distance peuvent être utilisés. La technique peut aussi être utilisés pour tester le trafic réseau. * Utiliser plusieurs clients physiques, par l’exécution de scripts de test pour charger le système.   Les tests de performance doivent être exécutés sur une machine dédiée ou à un moment dédié, afin de s’assurer de mesures contrôlées exactes.  Les bases de données utilisées pour les tests de performance doivent avoir une taille courante ou avoir une taille proportionnellement comparable. |

### Tests de charge

[Les tests de charges sont des tests de performance où les cibles de test sont soumis à différentes charges de travail afin d’évaluer leur performance et de s’assurer que leur fonction s’exécute normalement malgré les variations de charge. Le but des tests de charge est de déterminer et de s’assurer que le système fonctionne normalement par delà la charge de travail maximale. De plus, les tests de charge évaluent les caractéristiques de performance comme les temps de réponse, les taux de transaction et autres comportements pertinents.

NOTE : Les transactions sont, ci-dessous, des transaction d’affaire logiques. « Ces transactions sont définies comme des cas d’utilisation qu’un acteur du système est présumé exécuter en utilisant une cible de test comme, par exemple, ajouter ou modifier un contrat.]

|  |  |
| --- | --- |
| **Stratégie pour les tests de charge** | |
| Objectif de test: | Vérifier les temps de performance pour des transactions données ou pour des cas d’affaire selon différentes conditions de charge de travail. |
| Technique: | Utiliser des tests développer pour les tests fonctionnels ou du cycle d’affaire.  Modifier les fichiers de données pour accroître le nombre de transactions ou les tests pour accroître la fréquence des transactions. |
| Critère de complétion: | Une complétion est réussie lorsque des tests à plusieurs transactions et à plusieurs utilisateurs sans défaillance et dans un délai acceptable. |
| Considérations: particulières | Les tests de charges doivent être exécutés sur une machine dédiée, à un moment dédié afin de permettre un contrôle complet et des mesures exactes.  Les bases de données utilisées pour les tests de performance doivent avoir une taille courante ou avoir une taille proportionnellement comparable. |

### Tests de stress

[Les tests de stress sont un type de test de performance qui sont implémentés et exécutés pour trouver des erreurs imputables à un manque de ressources. Une anomalie de la cible de test peut être identifiée à cause d’une mémoire vive ou d’un espace disque insuffisants. D’autres anomalies peuvent être imputables à une concurrence d’accès à des ressources comme, par exemple, une largeur de bande réseau ou un accès concurrent à une base de données. Les stress tests peuvent aussi être utilisés pour identifier la pointe qu’une charge de travail peut atteindre.]

[NOTE : Les transactions sont, ci-dessous, des transaction d’affaire logiques.]

|  |  |
| --- | --- |
| **Stratégie pour les tests de stress** | |
| Objectif de test: | Vérifier que la cible de test fonctionne correctement et sans erreur dans les conditions de stress suivantes :   * Mémoire vive du serveur (RAM) ou unité de stockage à accès direct (DASD) disponible absente ou minime. * Nombre maximum de clients, actuels ou potentiels, connectés simultanément. * Plusieurs utilisateurs exécutent les mêmes transactions sur les mêmes données. * Le pire des cas en volume et en nature de transaction.   NOTES: Le but d’un test de stress doit aussi permettre d’identifier et de documenter les conditions sous lesquelles le système n’arrive pas à continuer de fonctionner correctement.  Les tests de stress pour la partie client sont décrits à la section 3.1.10 à la rubrique Tests de configuration. |
| Technique: | Utiliser des tests développés pour le profilage de la performance ou pour les tests de charge.  Pour tester un contexte de ressources limitées, les tests doivent être exécutés sur une seule machine où la RAM et le DASD doivent être réduits.  Pour les autres tests de stress, plusieurs clients doivent exécuter simultanément le même test ou des tests complémentaires pour simules l e pire des cas en volume et en nature de transaction. |
| Critère de complétion: | Tous les tests planifiés sont exécutés et les limites du système sont atteintes ou dépassées sans défaillance du logicielle ou lorsque la défaillance survient alors que les conditions ne sont pas les conditions spécifiées. |
| Considérations: particulières | Une surcharge de réseau peut requérir des outils réseau pour charger des messages ou des paquets.  Le DASD utilisé pour le système doit être temporairement réduit pour restreindre l’espace disque nécessaire à la croissance de la base de données.  Une synchronisation des accès simultanés des clients aux mêmes données est nécessaire. |

### Tests de volumétrie

[Les tests de volumétrie soumettent les cibles de test à d’important volume de données afin de déterminer quelles sont les limites du logiciel avant qu’il y ait défaillance. Les tests de volumétrie permettent aussi d’identifier la charge maximum ou le volume que les cibles de tests peuvent supporter pour une période de temps donnée. Ainsi, par exemple, pour la publication d’un rapport qui nécessite le traitement d’enregistrement de la base de données, le test de volumétrie devrait utiliser une base de données de grande taille et devra vérifier si le logiciel se comporte normalement et produit correctement le rapport.]

|  |  |
| --- | --- |
| **Stratégie pour les tests de volumétrie** | |
| Objectif de test: | Vérifier que la cible de test fonctionne correctement et sans erreur avec les scénarios de volume suivants :   * Nombre maximum de clients connectés qui exécutent la même fonction qui, en terme de performance, est le pire des cas pour une fonction d’affaire sur une longue période de temps. * Taille maximum de base de données et exécution simultanée de plusieurs transactions de requêtes ou rapports |
| Technique: | Utiliser des tests développés pour le profilage de la performance ou pour les tests de charge.  Plusieurs clients doivent exécuter simultanément le même test ou des tests complémentaires pour simules l e pire des cas en volume et en nature de transaction  Des bases de données de tailles différentes, courantes, mises à l’échelle ou chargées, sont créées et plusieurs clients exécutent des transactions de requêtes ou de rapport sur une longue période de temps. |
| Critère de complétion: | Tous les tests planifiés ont été exécutés et les limites spécifiées du système ont été atteintes ou dépassées sans défaillance logicielle. |
| Considérations: particulières | Définir une période de temps acceptable pour une exécution dans de conditions de volume élevé. |

### Tests de sécurité et de contrôle d’accès

[Les tests de sécurité et de contrôle d’accès portent sur deux domaines clés de la sécurité :

1. La sécurité au niveau de l’application incluant l’accès aux fonctions de traitement de données.
2. La sécurité au niveau du système, incluant la connexion à distance au système.

La sécurité au niveau de l’application contrôle l’accès des acteurs aux fonctions ou cas d’utilisation ou données qui leur sont rendues disponibles. Par exemple, tous les acteurs peuvent saisir des données et créer de nouveaux comptes, mais seuls les administrateurs peut les détruire. La sécurité au niveau des données est testée en s’assurant qu’un type d’utilisateur peut voir toutes les informations d’un client, incluant les données financières, alors qu’un autre type ne peut voir que les données démographiques pour le même client..

La sécurité au niveau du système permet aux seuls acteurs dont l’accès est autorisé, d’accéder aux applications par les seules passerelles appropriées.]

|  |  |
| --- | --- |
| **Stratégie pour les tests de sécurité et de contrôle d’accès** | |
| Objectif de test: | La sécurité au niveau de l’application: Vérifier qu’un acteur ne peut accéder aux seules données et fonctions pour lesquelles son type d’utilisateur lui donne les autorisations.  La sécurité au niveau du système : Vérifier que seuls les acteurs ayant la autorisation, ont un accès au système et aux application. |
| Technique: | Niveau application:   * Identifier et énumérer chaque type d’acteur avec les fonctions et données pour lesquelles il a les autorisations. * Créer des tests pour chaque type d’acteur afin et vérifier chaque autorisation en exécutant des transactions spécifique à chaque utilisateur acteur. * Modifier un type d’utilisateur et ré exécuter les tests pour le même utilisateur. Dans chaque cas, vérifier si les données et fonctions sont disponibles ou non disponibles tel que souhaité.   Niveau système: Voir les considérations particulières. |
| Critère de complétion: | Pour chaque type d’acteur connu, les fonctions et données appropriées sont disponibles et si toutes les transactions fonctionnent comme prévu et s’exécutent dans les précédents tests fonctionnels. |
| Considérations: particulières | L’accès au système doit être révisé et discuté avec les administrateurs appropriés de système ou de réseau. Ces tests peuvent ne pas être nécessaires s’ils sont déjà une fonction de l’administration du réseau ou du système. |

### Tests de basculement et de récupération

[Les tests de basculement et de récupération vérifient que la cible de test peut réussir à basculer et à récupérer pour différents dysfonctionnements matériels, logiciels ou de réseau sans qu’il y ait perte de données ou corruption de sonnées.

Les tests de basculement vérifient que, pour les systèmes qui doivent demeurer en opération, lorsqu’une défaillance survient, un système alternatif ou un système de sauvegarde prend le relais correctement sans perte de données ou de transactions.

Les tests de récupération est un test où le système est exposé à des conditions extrêmes, ou des conditions simulées pour provoquer une défaillance, telle qu’une panne d’un dispositif d’entrée-sortie ou des pointeurs de base de données invalides Les processus de récupération sont alors invoqués et l’application ou le système est alors suivi ou inspecté pour vérifier si la récupération des données, de l’application ou du système a été bien effectuée.]

|  |  |
| --- | --- |
| **Stratégie pour les tests de basculement et de récupération** | |
| Objectif de test: | Vérifier que le processus de récupération, manuel ou automatique, restaure correctement la base de données, les applications et le système dans un état connu désiré. Les types de conditions suivantes doivent être prises en compte :   * Mise hors tension du poste client. * Mise hors tension du serveur. * Interruption de communication du réseau * Mise hors tension ou de communication du DASD ou du contrôleur DASD. * Cycles incomplets: processus de filtrage de données interrompu, processus de synchronisation de données interrompu. * Pointeur ou clé de base de données invalide. * Élément de données de base de données corrompu ou invalide. |
| Technique: | Les tests créés pour les tests fonctionnels et du cycle d’affaire devraient être utilisés pour créer une série de transactions. Lorsque le moment désiré pour tester est atteint, les opérations suivantes devraient être exécutées ou simulées :   * Mise hors tension du poste client. * Mise hors tension du serveur ou initiation d’une procédure d’arrêt du serveur. * Interruption des serveurs réseau ou simulation de pertes de communication avec le réseau en débranchant physiquement ou en mettant hors tension les serveurs réseau ou les routeurs. * Interruption de communication ou mise hors tension du DASD ou du contrôleur DASD en simulant ou en éliminant physiquement la communication.   Lorsque ces opérations ont été exécutées, des transactions additionnelles devraient être exécutées jus qu’à un autre moment désiré pour tester soit atteint et on répète les opérations.  La même technique est utilisée pour des tests pour de cycles incomplets sauf que les processus de bases de données doivent être abandonner ou arrêtés prématurément.  À partir d’un état connu de base de données, corrompre des champs de données, des pointeurs et des clés de bas de données et exécuter des transactions en utilisant des tests fonctionnels ou des tests du cycle d’affaires. |
| Critère de complétion: | Dans tous les cas précédents, l’application, la base de données et le système devraient après les procédures de récupération retrouver un état connu et souhaité. Cet état comprend une corruption de données limitée aux champs de données, pointeurs et clés préalablement manipulés et des rapports indiquant que les processus de transaction incomplets ne sont pas dus aux interruptions. |
| Considérations: particulières | Les tests de récupération sont très intrusifs, Les procédures de débranchement peuvent ne pas être désirables ou faisables. Des méthodes alternatives comme des outils de diagnostic logiciels peuvent être nécessaires.  Des ressources système, de base de données ou du groupe réseau sont requises.  Les tests doivent être exécutés en dehors des heures ouvrables ou dans un environnement isolé. |

### Tests de configuration

[Les tests de configuration valident les opérations pour des cibles de test pour différentes configurations logicielles et matérielles. Les spécifications matérielles des postes de travail, des réseaux et des serveurs de base de données, en environnement de production, varient d’un client à l’autre. Les postes de travail peuvent avoir différents logiciels en opération selon différentes combinaisons et utilisant différentes ressources.]

|  |  |
| --- | --- |
| **Stratégie pour les tests de configuration** | |
| Objectif de test: | Vérifier que la cible de test fonctionne correctement avec les configurations logicielle et matérielle définies. |
| Technique: | Utiliser des scripts de tests fonctionnels  Ouvrir et fermer des logiciels qui ne sont pas des logiciels liés à la cible de tests, comme Microsoft Excel ou Microsoft Word, avant de démarrer le test.  Exécuter des transactions afin de simuler des acteurs qui interagissent avec la cible de test et avec un logiciel qui n’est pas associé à la cible de tests.  Répéter le processus en réduisant la mémoire conventionnelle disponible sur le poste de travail. |
| Critère de complétion: | Our chaque combinaison de cible de test et de logiciel non lié à la cible de test, toutes les transactions doivent être complétées sans défaillance. |
| Considérations: particulières | Déterminer quels sont les logiciels non liés à la cible de test disponible et accessible sur le bureau.  Déterminer quelles sont les applications habituellement utilisées.  Déterminer les données traitées par ces applications comme, par exemple, un document Word de 100 pages.  Tout le système, le réseau, les serveurs réseau, les bases de données, etc. devraient être aussi documentées lors de l’exécution de ces tests. |

### Tests d’installation

[Les tests d’installation poursuivent deux objectifs. D’abord, s’assurer que le logiciel peut être installé dans différentes conditions, tant pour les nouvelles installations, les mises à jour que pour compléter une installation personnalisée dans des conditions normales et des conditions anormales. Les conditions anormales comprennent un espace disque insuffisant, des autorisations insuffisantes pour créer des répertoires, etc. L’autre objectif est de vérifier que, une fois installé, le logiciel fonctionne correctement. Cela signifie qu’il faut exécuter un certain nombre de tests fonctionnels.]

|  |  |
| --- | --- |
| **Stratégie pour les tests d’installation** | |
| Objectif de test: | Vérifier que la cible de test est installée correctement pour chaque configuration matérielle requise, avec les conditions suivantes :   * Nouvelle installation, sur une nouvelle machine où l’application n’a jamais été installée auparavant. * Mise à niveau d’une machine avec une application de la même version. * Mise à jour d’une machine avec une application d’une plus vieille version. |
| Technique: | Valider les conditions de la machine cible manuellement ou avec des scripts automatiques : nouvelle installation, réinstallation de la même version ou réinstallation d’une vieille version.  Exécuter une installation.  Utiliser un sous-ensemble de scripts de tests fonctionnels et exécuter les transactions. |
| Critère de complétion: | Les transactions sont exécutées avec succès sans défaillance. |
| Considérations: particulières | Déterminer quelles sont les transactions qui garantiront que l’application sera installée avec succès sans que des composants majeurs ne soient oubliés. |

## Outils

Les outils suivants seront employés pour ce projet:

[NOTE: Supprimer ou ajouter les items selon le cas.]

|  |  |
| --- | --- |
|  | **Outil** |
| Gestion des tests |  |
| Suivi des anomalies |  |
| Tests fonctionnels |  |
| Tests de performance |  |
| Gestion de projet |  |
|  |  |
|  |  |

# Ressources

[Énumérer les ressources nécessaires à l’effort de test, leurs principales responsabilités, leur connaissance et leurs qualifications.]

## Travailleurs

Ce tableau énumère les ressources humaines nécessaires à la réalisation de ce plan.

[NOTE: Supprimer ou ajouter les items selon le cas.]

|  |  |  |
| --- | --- | --- |
| **Ressources humaines** | | |
| **Travailleur** | **Nombre minimum (plein temps)** | **Responsabilités - commentaires** |
| Responsable des tests |  | Fournir un encadrement de gestion  Responsabilités:   * Fournir une direction technique * Recruter les ressources appropriées * Gérer les rapports |
| Concepteur de test |  | Identifier, prioriser et implémenter les cas de test  Responsabilités:   * Générer le plan de tests * Générer le modèle de tests * Évaluer l’efficacité de l’effort de test |
| Testeur |  | Exécuter les tests  Responsabilités:   * Exécuter les tests * Journaliser les résultats * Reprendre sur les erreurs * Documenter les anomalies et les demandes de changement. |
| Administrateur de tests système |  | Mettre en place et maintenir un environnement de tests  Responsabilités:   * Administrer le système de gestion de tests * Installer et gérer l’accès des travailleurs au système de tests. |
| Administrateur de tests de base de données |  | Mettre en place et maintenir les données de tests.  Responsabilités:   * Administrer les bases de données de tests |
| Concepteur |  | Identifier et définir les opérations, attributs et associations des classes test.  Responsabilités:   * Identifier et définir les classes de test. * Identifier et définir les paquetages de test |
| Implémenteur |  | Implémenter et exécuter les tests unitaires de classes de test et de paquetages de test.  Responsabilités:   * Créer des classes de test et les paquetages de test du modèle de test. |

## Système

Le tableau suivant identifie les ressources du système nécessaire à l’exécution du plan de tests.

Les éléments du système de tests ne sont pas tous connus à ce jour. Il est recommandé que le système simule un environnement de production en réduisant les accès et la taille des bases de données lorsque nécessaire.

[NOTE: Supprimer ou ajouter les items selon le cas.]

|  |  |
| --- | --- |
| Ressources du système | |
| Ressource | Nom - Type |
| Serveur de base de données |  |
| * Réseau – sous réseau | TBD |
| * Nom du serveur | TBD |
| * Nom de la base de données | TBD |
| Poste de travail |  |
| * Exigences de configuration spéciales | TBD |
| Référentiel de tests |  |
| * Réseau – sous réseau | TBD |
| * Nom du serveur | TBD |
| Poste de développement de tests | TBD |

# Jalons du projet

[L’exécution des tests doit inclure les activités de test pour chaque effort de test identifié dans les sections précédentes. Des jalons distincts devraient être identifiés pour transmettre le statut du projet de tests et ses réalisations.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Activité** | **Effort** | **Début** | **Fin** |
| Planification des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Conception des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Implémentation des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Exécution des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Évaluation des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |

# Biens livrables

[Énumérer les artéfacts, documents, outils et rapports qui seront créés, leur auteur, leur destinataire et la date de leur livraison]

## Modèle de test

[Identifier les rapports qui seront créés et extraits du modèle de tests.]

## Journaux de test

[Décrire la méthode et les outils utilisés pour enregistrer et faire rapport des résultats de test et du statut des tests.]

## Rapports d’anomalies

[Identifier la méthode et les outils utilisés pour enregistrer, tracer et faire rapport des incidents de test et de leur statut.]

# Annexe A: Tâches du projet

Voici la liste des tâches liées à l’exécution des tests :

* Planifier les tests
  + Identifier les exigences de test
  + Évaluer les risques
  + Identifier les ressources nécessaires aux tests
  + Développer une stratégie de test
  + Créer le calendrier de tests
  + Générer le plan de test
* Concevoir les tests
  + Analyser la charge de travail
  + Identifier et décrire les cas de test
  + Identifier et structurer les procédures de test
  + Réviser la couverture des tests
* Implémenter les tests
  + Enregistrer ou programmer les scripts de test
  + Identifier les fonctionnalités propres au test dans les modèles de conception et d’implémentation.
  + Établir des ensembles de données externes
* Exécuter les tests
  + Exécuter les procédures de test
  + Évaluer l’exécution des tests
  + Récupérer les tests suspendus
  + Vérifier les résultats
  + Examiner les résultats non prévus
  + Enregistrer les anomalies
* Évaluer les tests
  + Évaluer la couverture des cas de tests
  + Évaluer la couverture du code
  + Analyser les anomalies
  + Déterminer si les critères de complétion et de succès des tests ont été atteints.