LOG430-01-Equipe 02

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Tester l’application avec un couvrage de 95% des lignes de codes ainsi qu’une mise en forme standardisé du code. | |
| Détails du scénario | Stimulus | Le code en train d’être poussé sur le répertoire git. |
| Source du stimulus | Le script de test automatique |
| Environnement | Développement |
| Artéfact | Tous le code testé par les tests unitaire |
| Réponse | Rapport de test et rapport de la mise en page du code (LINT) |
| Mesure de réponse | * Couvrage : % * Temps : MS * Fautes : Par ligne de code * Status des tests : Passe ou échec |
| Décisions architecturales et raisonnement | * Un hook est fait après un push GIT * Réutiliser du code existant car celui-ci suit déjà les standards établis * Contrôle de l’état du système : * **Exécutable assertion au niveau du processus** : le système de mise en forme du code bloque le git push | |
| Risques | * Les tests ne passent pas * Laise en forme (Lint) n’est pas respecté au niveau du code. | |
| Compromis |  | |

LOG430-01-Equipe 04

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Tests unitaires sur l’agrégateur de données | |
| Détails du scénario | Stimulus | L’exécution des tests |
| Source du stimulus | le développeur de test |
| Environnement | Environnement de développement |
| Artéfact | Agrégateur de données |
| Réponse | Les tests sont exécutés |
| Mesure de réponse | Plus de 95% de couverture |
| Décisions architecturales et raisonnement | Utilisation d’une base de données de tests.  Séparation des concepts de l’agrégateur et du subscriber afin de séparer ces services et de faciliter la testabilité de l’agrégateur. | |
| Risques | Aucun | |
| Compromis | ~~Temps de développement pour créer une base de données de test.~~ | |

LOG430-01-Equipe 05

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Exécuter des tests unitaires qui vérifient la conformité des données. | |
| Détails du scénario | Stimulus | Exécution des tests unitaires suite à l’implémentation de l'agrégateur. |
| Source du stimulus | Développeur interne |
| Environnement | Environnement de test qui comprend un fichier de données de tests. |
| Artéfact | Agrégateur de données |
| Réponse | Exécuter la suite de tests et capturer les résultats. |
| Mesure de réponse | - Nombre de tests réussis  - Pourcentage de couverture d’au moins 95% en une minute. |
| Décisions architecturales et raisonnement | Prévoir un fichier de test | |
| Risques | Le fichier de test n’est pas représentatif du fichier de données normal. | |
| Compromis | Le fichier de test ne nous permet pas de prendre en compte le côté performance de l’agrégateur, il ne nous permet de vérifier que les résultats. | |

LOG430-01-Equipe 07

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Exécuter les tests unitaires qui génèrent une liste des accidents nécessitant les données du serveur de la Ville. | |
| Détails du scénario | Stimulus | Séries de tests unitaires automatisés |
| Source du stimulus | Développeur interne |
| Environnement | Environnement de développement |
| Artéfact | Sous-système qui transforme les données de la Ville en paramètres pour notre application |
| Réponse | Résultats de la séquences des tests |
| Mesure de réponse | Couvrage de 95% des tests de la séquence |
| Décisions architecturales et raisonnement | * Mocker les données provenant de l’API de la Ville. * Écrire des tests unitaires en fonction des tâches décrites dans les autres cas d’utilisation. (Par ex. CU01-Lister les accidents: Tester qu’il est possible de dresser la liste à l’aide des données simulées de la Ville. | |
| Risques | Aucun contrôle sur la récupération des données de la Ville, puisqu’on simule leur réponse. | |
| Compromis | Les tests qui nécessitent une connection à l’API de la Ville peuvent toujours être exécutés lorsque ce dernier est hors service. | |

LOG430-01-Equipe 09

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Exécuter des tests unitaires qui génèrent des données en lien avec la qualité de l’air. | |
| Détails du scénario | Stimulus | Lancement des tests unitaires |
| Source du stimulus | Développeur ou Testeur interne. |
| Environnement | Environnement de développement isolé. |
| Artéfact | Sous-système d’agrégation de données. |
| Réponse | Résultats des tests, positif ou négatif. |
| Mesure de réponse | Couverture de 95%. |
| Décisions architecturales et raisonnement | - Utilisation de fausses données invalides de la ville pour simuler un mal fonctionnement.  - Tests unitaire qui couvre l’ensemble des cas d’utilisation. | |
| Risques | - Les erreurs non-géré peuvent avoir un impact sur la disponibilité du service si le système tombe en état d’erreur. | |
| Compromis | - Disponibilité | |

LOG430-01-Equipe 10

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Exécuter des tests unitaires sur les routes de BoraREST | |
| Détails du scénario | Stimulus | Séries de tests unitaires |
| Source du stimulus | Unit testers (Développeur de l’API) |
| Environnement | Environnement de test |
| Artéfact | Le code de l’API BocaREST |
| Réponse | Exécution des tests et capture des résultats |
| Mesure de réponse | Le nombre de tests échoués  Le temps d'exécution des tests  95% pourcentage de ligne couvert |
| Décisions architecturales et raisonnement | * Sandbox : Permettre d’avoir un contrôle sur les donnés et de ne pas affecté les données récoltées dans la base de données. * Executable assertion | |
| Risques |  | |
| Compromis |  | |

LOG430-01-Equipe 11

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité |  | |
| Scénario / cas d’utilisation |  | |
| Détails du scénario | Stimulus | Le générateur de données de traffic moyen entre deux arrêt est terminé |
| Source du stimulus | Les test unitaires |
| Environnement | Développement |
| Artéfact | Le module est testé |
| Réponse | Le rapport de test |
| Mesure de réponse |  |
| Décisions architecturales et raisonnement |  | |
| Risques |  | |
| Compromis |  | |

LOG430-01-Equipe 20

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Exécution des tests unitaires | |
| Détails du scénario | Stimulus | implémentation d’une fonctionnalité |
| Source du stimulus | ~~testeur~~ développeur |
| Environnement | Développement |
| Artéfact | serveur node.js |
| Réponse | exécution des tests unitaires |
| Mesure de réponse | couverture du code à 95% |
| Décisions architecturales et raisonnement | assertions exécutable afin de faciliter les test | |
| Risques | chance de faire crasher le programme si une exception est lancé mais pas géré | |
| Compromis | disponibilité et performance | |

LOG430-01-Equipe 22

|  |  |  |
| --- | --- | --- |
| Analyse détaillée de scénario | | |
| Attribut de qualité | Testabilité | |
| Scénario / cas d’utilisation | Tester notre système reçoit bien les données de la ville de Montréal | |
| Détails du scénario | Stimulus | Testeur unitaire |
| Source du stimulus | Complétion du codage pour recevoir les donnés de la ville de Montréal |
| Environnement | Lors du développement |
| Artéfact | Module de traitement de notre système |
| Réponse | Exécute les tests et analyse les résultats |
| Mesure de réponse | Couverture du code doit atteindre 80% |
| Décisions architecturales et raisonnement | Limiter le structure de notre système pour tester seulement la partie de la réception de données | |
| Risques | Aucun risque | |
| Compromis | tester les valeurs extrêmes au lieu de faire tous les cas possible | |