
Équipe 03

Cégep à distance
Plan de tests logiciels
Version 1.0

Historique des révisions

Date	Version	Description	Auteur
2022-01-26	1.0	Rédaction initiale du document	Laura Beaudoin, Sandu Babira, Jasmine Roy, Guillaume Rompré, Anoir Dhaoui

Table des matières

1. Introduction	4
2. Exigences à tester	4
3. Stratégie de test	5
3.1. Types de test	5
3.1.1. Tests de fonction	5
3.1.2. Tests d'interface usager	6
3.1.3. Tests d'intégrité des données	6
3.1.5. Tests de charge	6
3.1.6. Tests de stress	7
3.1.7. Tests de volume	7
3.1.8. Tests de sécurité et de contrôle d'accès	7
3.1.9. Tests d'échec/récupération	8
3.1.10. Tests de configuration	8
3.2. Outils	8
4. Ressources	9
4.1. Équipe de test	9
4.2. Système	9
5. Jalons du projet	9

Plan de tests logiciels

1. Introduction

Ce document présente le plan de tests pour l'application du Tableau de bord du Cégep à distance. On y spécifie dans un premier lieu les exigences (décrites dans le SRS) qui seront soumises à la discipline de tests ainsi que les types de tests choisis pour chacune. Ensuite, la stratégie de test sera présentée, en plus d'une brève description de chaque type de test associé aux exigences et les outils nécessaires pour ces derniers. Finalement, les ressources nécessaires et les jalons prévus pour cette discipline seront présentés.

2. Exigences à tester

Tableau 1 : Liste des exigences soumises à la discipline de tests et les types de tests associés.

Exigences (# SRS)	Type de test
3.1.1 Gestion de comptes utilisateurs	Tests de fonction, d'interface usager
3.1.1.1 Création de comptes étudiants par le système	Tests d'intégrité de données
3.1.1.2 Définition d'un mot de passe par l'utilisateur	Tests d'intégrité de données
3.1.1.3 Connexion de l'utilisateur	Tests de sécurité et de contrôle d'accès
3.1.2 Notifications par courriel	Tests de fonction
3.1.2.6.1 Changement du mot de passe à partir du lien de courriel	Tests d'intégrité de données
3.1.3 Questionnaires	Tests de fonction, d'interface usager
3.1.3.1 Envoyer un questionnaire complété	Tests d'intégrité de données
3.1.3.7 Continuer à remplir un questionnaire partiellement complété	Tests d'intégrité de données
3.1.4 Roue de la réussite	Tests de fonction, d'interface usager
3.1.5 Parcours temporel	Tests de fonction, d'interface usager
3.1.6 Prédiction d'abandon	Tests de fonction
3.1.7 Outils administratifs	Tests de fonction, d'interface usager, de sécurité et de contrôle d'accès
3.1.8 Soutien technique	Tests de fonction, d'interface usager
3.2 Exigences souhaitables	Tests de fonction
3.2.1 Gestion de compte observateur	Tests sécurité et de contrôle d'accès
3.2.3 Outils administratifs	Tests sécurité et de contrôle d'accès
4.1.2 Navigateurs	Tests de configuration

Exigences (# SRS)	Type de test
4.2.1 Disponibilité	Tests d'échec/récupération
4.3 Performance	Tests de charge, de stress et de volume

3. Stratégie de test

Pour assurer le développement d'un produit conforme aux normes et aux exigences établies dans le SRS, il est prévu d'intégrer les tests de fonction au processus de développement logiciel. Ainsi, la fin d'une tâche d'implémentation de fonctionnalité sera accompagnée d'une période de test (en boîte noire). Ces derniers seront principalement des tests manuels visant à vérifier l'adéquation fonctionnelle du code implémenté. Les fonctionnalités plus critiques (surtout celles reliées à la base de données) seront soumises à des tests plus rigoureux, incluant des tests d'intégrité des données, afin de minimiser le risque d'erreurs. L'équipe désire ainsi s'assurer qu'après chaque étape d'implémentation un livrable entièrement fonctionnel et conforme aux exigences soit produit.

De plus, dans les étapes finales de production, les efforts des tests se concentreront sur les interfaces utilisateur, ainsi que la performance du système. Par conséquent, des tests d'utilisation, de charge, de stress et de volume seront de mise.

3.1. Types de test

3.1.1. Tests de fonction

Tableau 2 : Objectif, technique et critères de complétion des tests de fonction.

Objectif de test:	Valider que la fonctionnalité du système respecte le cas d'utilisation prévu.
Technique:	Utiliser l'application comme elle est prévue. Le système est parcouru comme il le serait dans une utilisation normale par un utilisateur lambda.
Critère de complétion:	Le comportement du système est conforme aux exigences et le système ne présente aucun message d'erreur ou de défaillance.
Considérations spéciales:	Ne pas tester de comportement inhabituel. Les cas d'utilisations à tester doivent être conformes aux exigences du SRS.

3.1.2. Tests d'interface usager

Tableau 3 : Objectif, technique et critères de complétion des tests d'interface usager.

Objectif de test:	Valider l'intuitivité et l'accessibilité de l'interface de l'application. Valider que l'interface est adaptée aux utilisateurs cibles.
Technique:	Présenter l'application à des utilisateurs externes (ne faisant ni partie de l'équipe de développement ni de l'équipe du client ayant travaillé avec l'équipe de développement), leur donner une liste d'actions à effectuer sans aide ou intervention et prendre note de leurs commentaires et de leurs difficultés.
Critère de complétion:	L'utilisateur réussit à effectuer les actions sans recourir à l'aide extérieure.
Considérations spéciales:	Aucune.

3.1.3. Tests d'intégrité des données

Tableau 4 : Objectif, technique et critères de complétion des tests d'intégrité des données.

Objectif de test:	Assurer l'intégrité des données même si des actions inhabituelles surviennent et si des données incompatibles sont envoyées par erreur dans la base de données.
Technique:	Insérer des données incompatibles dans la base de données à l'aide de requêtes manuelles. Supprimer manuellement des entrées ayant des dépendances.
Critère de complétion:	Les valeurs impossibles doivent être refusées par la base de données. Les entrées supprimées ayant des dépendances doivent déclencher une mise à jour en cascade des tables qui en dépendent ou être bloquées.
Considérations spéciales:	Aucune.

3.1.5. Tests de charge

Tableau 5 : Objectif, technique et critères de complétion des tests de charge.

Objectif de test:	Valider la capacité du système à maintenir les fonctionnalités selon le niveau de performance attendu malgré une demande accrue pour les charges suivantes : • Des étudiants remplissent et soumettent le questionnaire de parcours (exigences 3.1.3.1 et 3.1.3.10) • Des administrateurs testent l'algorithme de parcours (exigences 3.1.7.2)
Technique:	Plusieurs utilisateurs concurrents effectuent les charges de travail (<i>workload</i>) prédéfinies. Chronométrer le temps de chargement des pages.
Critère de complétion:	Le système fonctionne sans faire défaut. La performance respecte les exigences spécifiées à la section 4.3 du SRS.
Considérations spéciales:	Ne pas tester de comportement inhabituel. L'application doit être utilisée tel que prévu.

3.1.6. Tests de stress

Tableau 6 : Objectif, technique et critères de complétion des tests de stress.

Objectif de test:	Valider le fonctionnement adéquat du système, lorsque soumis à un grand nombre d'utilisateurs simultanés.
Technique:	Simuler le cas où un grand nombre d'utilisateurs se connectent et utilisent l'application simultanément.
Critère de complétion:	Les utilisateurs peuvent utiliser l'application sans défaillance. La performance respecte les exigences spécifiées à la section 4.3 du SRS.
Considérations spéciales:	La charge dans ce test sera considérablement supérieure au cas d'usage typique (le but étant d'explorer les limites du système).

3.1.7. Tests de volume

Tableau 7 : Objectif, technique et critères de complétion des tests de volume.

Objectif de test:	Valider le fonctionnement adéquat du système, lorsque soumis à une quantité de données à traiter hors norme.
Technique:	Imposer au serveur le traitement d'une quantité hors norme d'inscriptions lors d'une mise à jour quotidienne.
Critère de complétion:	Le système ne doit pas présenter de défaillance. Le serveur doit traiter toutes les inscriptions sans causer de panne.
Considérations spéciales:	Aucune.

3.1.8. Tests de sécurité et de contrôle d'accès

Tableau 8 : Objectif, technique et critères de complétion des tests de sécurité et de contrôle d'accès.

Objectif de test:	Valider l'incapacité d'un utilisateur à se connecter avec des informations erronées ou dans un mauvais type de compte. Valider l'accès aux fonctionnalités des différents types de comptes.
Technique:	Tenter de se connecter avec des informations erronées. Tenter de se connecter avec des <i>informations non concordantes</i> . Tenter de connecter un compte utilisateur comme un autre type de compte. Tenter d'accéder aux outils administrateurs avec un compte étudiant. Tenter d'accéder à une fonctionnalité privée sans se connecter. Tenter de créer deux comptes avec le même mot de passe et s'assurer que les <i>hash</i> sont différents.
Critère de complétion:	Les connexions et accès inadéquats sont bloqués.
Considérations spéciales:	Aucune.

3.1.9. Tests d'échec/récupération

Tableau 9 : Objectif, technique et critères de complétion des tests d'échec/récupération.

Objectif de test:	Valider la capacité du système à préserver l'intégrité des données et à se rétablir à la suite d'une panne.
Technique:	Fermer abruptement le serveur alors que des utilisateurs sont connectés. Déconnecter les utilisateurs de façon inadéquate au milieu d'une tâche en coupant l'accès internet, utilisant le gestionnaire des tâches de l'ordinateur pour en couper le processus, etc.
Critère de complétion:	Les utilisateurs reçoivent un message d'erreur s'ils tentent d'effectuer de nouvelles actions. La base de données ne présente pas de corruption. Les dernières actions enregistrées de l'utilisateur sont accessibles à la prochaine connexion.
Considérations spéciales:	Aucune

3.1.10. Tests de configuration

Tableau 10 : Objectif, technique et critères de complétion des tests de configuration.

Objectif de test:	Valider le fonctionnement du système sur différents navigateurs présentés dans la section 4.1.2 du SRS.
Technique:	Ouvrir le navigateur à tester. Accéder à l'application web à partir du navigateur. Utiliser normalement l'application.
Critère de complétion:	Le fonctionnement du système est conforme aux exigences définies dans le SRS.
Considérations spéciales:	Ne pas tester de comportement inhabituel. L'application doit être utilisée tel que prévu.

3.2. Outils

Les outils suivants seront utilisés au sein de la discipline de test:

Tableau 11 : Les outils utilisés lors de la discipline de test.

Type de test	Outil
Test de charge, de stress et de volume	JMeter (+chronomètre) ou Loadmill
Test de fonction, d'interface usager, configuration	Ordinateur et cellulaire

4. Ressources

4.1. Équipe de test

Tableau 12 : Les membres de l'équipe de test, leurs rôles et responsabilités.

Rôle	Membre de l'équipe	Responsabilités
Testeur, Correcteur	Guillaume Rompré	Conception et application de tests sur le module IA de prédiction des abandons Correction des erreurs identifiées
Testeur, Correcteur	Sandu Babira	Conception et application de tests sur le serveur backend du Tableau de bord Correction des erreurs identifiées
Testeur, Correcteur	Jasmine Roy	Conception et application de tests sur le serveur frontend du Tableau de bord Correction des erreurs identifiées

4.2. Système

Les systèmes nécessaires à l'exécution des tests devront correspondre le plus possible à l'environnement de déploiement de l'application, surtout lors de l'application des tests de performance. En effet, afin d'évaluer le comportement du système le plus exactement possible, l'environnement de test devrait émuler la situation normale d'exécution du système. L'exécution des tests manuels ne nécessite que d'une machine capable de rouler un fureteur web (idéalement tous les types identifiés dans les exigences du SRS).

5. Jalons du projet

Tableau 13 : Les jalons reliés à la discipline de test, l'effort estimé, ainsi que leurs dates de début et de fin.

Jalon	Effort	Date de début	Date de fin
Programmation des interfaces utilisateur et logique de gestion des comptes étudiant Conception et programmation des modèles de prédiction des abandons	35h	7 février 2022	27 février 2022
Programmation des questionnaires et logique de calcul des résultats Déploiement des modèles de prédiction des abandons et connexion avec les bases de données	35h	28 février 2022	20 mars 2022
Implémentation des comptes alternatifs et outils administratifs	20h	21 mars 2022	3 avril 2022
Tests de fin de projet	35h	4 avril 2022	19 avril 2022