

***Fais-moi un dessin***  
**Plan de tests logiciels**

**Version 1.4**

## Historique des révisions

Date	Version	Description	Auteur
2020-04-09	1.0	sections 1, 3.1.1	Geneviève Laroche
2020-04-11	1.2	sections 2, 3.1.2, 3.1.3, 3.1.5, 4.1, 5	Geneviève Laroche
2020-04-13	1.3	sections 2, 3.1.4, 4.2	Geneviève Laroche
2020-04-13	1.4	Révision	Annie Rochette

# Table des matières

<b>1. Introduction</b>	<b>4</b>
<b>2. Exigences à tester</b>	<b>4</b>
<b>3. Stratégie de test</b>	<b>4</b>
3.1. Types de test	4
3.1.1. Tests de fonction	4
3.1.2. Tests d'interface usager	5
3.1.3. Tests d'intégrité des données	5
3.1.4. Tests de charge	5
3.1.5. Tests de volume	6
3.2. Outils	6
<b>4. Ressources</b>	<b>6</b>
4.1. Équipe de test	6
4.2. Système	7
<b>5. Jalons du projet</b>	<b>7</b>

# Plan de tests logiciels

## 1. Introduction

Ce document de plan de tests logiciels pour l'application *Fais-moi un dessin* a comme objectifs d'identifier les exigences devant être testées, d'établir une stratégie de tests ainsi que de déterminer les ressources à allouer et les principaux jalons en lien avec les tests tout au long de la réalisation du projet.

## 2. Exigences à tester

Exigences	Tests associés
3.1.2.1 Création d'un nouveau compte utilisateur	Tests de fonction
3.1.2.2, 4.3.1 Connection au site	Tests de fonction et de charge
3.1.1 Clavardage	Tests de volume
4.1.1. Compréhensibilité de l'interface utilisateur	Tests d'interface usager
Les requêtes HTTP doivent être fonctionnelles	Tests de fonctions, test d'intégrité des données
Intégrité des données de la base de donnée	Tests d'intégrité des données

## 3. Stratégie de test

### 3.1. Types de test

#### 3.1.1. Tests de fonction

Objectif de test:	Ce type de test vise à s'assurer que le résultat d'une fonction est conforme aux attentes selon son cas d'utilisation.
Technique:	Exécuter la fonctionnalité et vérifier que le résultat est bien conforme aux attentes selon ce qui a été convenu dans le SRS et dans les cas d'utilisation.
Critère de complétion:	L'exécution de la fonctionnalité mène aux résultats attendus même lors de cas limites et possiblement problématiques.
Considérations spéciales:	Si applicable, exécuter la fonctionnalité de façon abusive afin de tester les cas limites.

### 3.1.2. Tests d'interface usager

Objectif de test:	Ce type de test vise à s'assurer qu l'interface utilisateur est compréhensible.
Technique:	Présenter l'interface à des personnes qui ne travaillent pas sur le projet pour qu'elles se promènent à travers l'interface. Ensuite, nous notons leurs commentaires, leurs questionnements et leurs difficultés rencontrés..
Critère de complétion:	Les ressources externes sont capables d'utiliser l'application sans trop de difficultés, leurs commentaires sont généralement positifs.
Considérations spéciales:	Les ressources externes sélectionnées doivent faire partie du public cible et avoir une compréhension minimale du fonctionnement d'un ordinateur ou d'une tablette.

### 3.1.3. Tests d'intégrité des données

Objectif de test:	Ce type de test vise à s'assurer que la base de données n'accepte que des valeurs permises.
Technique:	Envoi de requêtes valides et invalides avec Postman.
Critère de complétion:	Les requêtes invalides ne se rendent pas à la base de données et les requêtes valides ne rencontrent aucune erreur.
Considérations spéciales:	N/A

### 3.1.4. Tests de charge

Objectif de test:	Ce test vise à s'assurer du bon fonctionnement de l'application lorsque de nombreuses personnes l'utilisent en même temps.
Technique:	Quatre personnes ou plus tentent de se connecter au même moment exact afin d'observer si l'application cesse de fonctionner adéquatement.
Critère de complétion:	L'application n'est pas significativement ralentie et ne 'crash' pas.
Considérations spéciales:	N/A

### 3.1.5. Tests de volume

Objectif de test:	Ce test vise à s'assurer que l'application est en mesure de recevoir de grosses quantités de données sans échouer ni être ralentie.
Technique:	Envoyer beaucoup d'informations, très rapidement au système et observer comment il réagit.
Critère de complétion:	L'application continue de fonctionner comme elle se doit et n'est pas ralentie.
Considérations spéciales:	N/A

## 3.2. Outils

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

Type de test	Outil
Tests de fonction	Visual Studio, Visual Studio Code
Tests d'interface usager	Un utilisateur (ressource externe)
Tests d'intégrité des données	Postman
Tests de charge	Aucun outil particulier. Plusieurs utilisateurs doivent se connecter en même temps avec leurs PC/tablettes.
Tests de volume	Aucun outil particulier. Au moins un utilisateur doit envoyer un nombre excessif de requêtes.

## 4. Ressources

### 4.1. Équipe de test

Rôle	Membre de l'équipe	Responsabilités
Conceptrice de tests côté client lourd	Geneviève Laroche	Rédaction du plan de test, identification et rédaction de cas de tests pour le client lourd.
Testeur côté client lourd	Manyl Telmat	Exécuter les tests, noter les résultats et les modifications nécessaires.

Concepteur & testeur côté serveur	Maxime Bernier	Tester les requêtes du serveur sur Postman, conception et exécution de tests unitaires.
Conceptrice de tests côté client léger	Annie Rochette	Identification et rédaction de cas de tests pour le client léger.
Testeuse côté client léger	Florence Cimon-Paquet	Exécuter les tests, noter les résultats et les modifications nécessaires.

#### 4.2. Système

Aucune ressource système particulière sera nécessaire.

#### 5. Jalons du projet

Jalon	Effort	Date de début	Date de fin
Planification des tests	2	10/01/01	26/01/01
Conception des tests	3	28/01/01	12/02/01
Implémentation des tests	3	12/02/01	20/02/01
Exécution des tests	6	22/02/01	20/03/01
Évaluation des tests	2	24/03/01	30/03/01