

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

Editeur d'automates: “AutomatesLab”

Stratégie de tests

Référence : ST-8-0B
Fournisseur :
Date : 08/01/2023
Version/Édition : 0A
État : Préliminaire

Type de diffusion : Diffusion propriétaire
Autre référence :

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

Historique des révisions

Date	Description et justification de la modification	Pages / Chapitre	Edition / Révision
08/01/2023	Création	Toutes	0A
20/01/2023	Ajout 1.3, 4.1 IHM, et corrections mineurs	Toutes	0B

Ref : ST-8-0A	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates		

Table des matières

1. Introduction	4
1.1. Objectif du document	4
1.2. Présentation du logiciel	5
1.3. Hypothèses et risques	6
1.4. Glossaire et abréviations	6
2. Organisation de la validation	8
2.1. Documents de tests	8
2.2. Environnement de test	8
3. Ressources	9
3.1. Ressources humaines	9
3.2. Ressources matérielles	9
4. Stratégie de test	10
4.1. Stratégie de validation	10
4.1.1. Principe	10
4.1.2. Organisation de la stratégie	10
4.1.3. Objectif de test	10
4.1.3.1. Validation des règles de gestion	11
4.1.3.2. Validation des règles de calcul	11
4.1.3.3. Validation des règles d'enchaînement	11
4.1.3.4. Validation des procédures d'installation et d'exploitation	11
4.1.4. Niveaux de criticité	12
4.1.5. Critères d'arrêt des tests	12
4.1.5.1. Normal	12
4.1.5.2. Anormal	12
5. Description des tests	13
5.1. Validation des projets	13
5.1.1. Dossier de tests unitaire	13
5.1.1.1. Correspondance	13
5.1.2. Dossier de tests de validation	13
5.1.2.1. Simulation des utilisateurs	13
5.1.2.2. Correspondance	13
5.2. Matrice de couverture	14

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

1. Introduction

1.1. Objectif du document

Ce document a pour objectif de décrire la méthodologie Fournisseur concernant le test et la validation d'application. Il aborde successivement l'organisation, la planification des tests et la préparation des campagnes de tests à effectuer.

Il aborde les points suivants:

- l'organisation du projet de test validation, rôles et responsabilités des intervenants dans le processus de test,
- la communication entre les intervenants responsables des tâches à effectuer ainsi que leur planification,
- les activités de préparation et de conduite des tests,
- la stratégie de test de validation appliquée au projet,
- les outils et les environnements dans lesquels les tests doivent s'exécuter,
- les niveaux de criticité et les critères d'arrêt des tests,
- le cycle de prise en compte des anomalies rencontrées dans l'exécution des tests ainsi que leur traitement,
- le mode d'archivage des éléments issus des étapes de test.

La définition de ces activités doit permettre d'automatiser les campagnes de tests afin d'obtenir les éléments indispensables à l'exécution des tests de non régression lors des phases d'évolutions et de corrections de l'application.

Ce document est applicable sur les modules entrant dans le cadre du développement de l'application à développer.

Ref : ST-8-0A	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates		

1.2. Présentation du logiciel

Le projet Editeur d'Automate a pour objectif de créer une application (appelée AutomatesLab) permettant de définir des automates finis et de les sauvegarder dans un format XML adapté. Un automate peut être chargé dans l'éditeur depuis son fichier XML.

AutomatesLab est une application permettant d'éditer des automates de manière graphique. Un automate est un objet mathématique très utilisé dans de nombreux domaines de l'informatique. Il se compose d'un ensemble d'états et de transitions qui, ensemble, permettent de reconnaître les mots d'un langage composé d'un ensemble de caractères fini.

L'interface du logiciel se veut simple et intuitive et est agencée de la façon suivante :

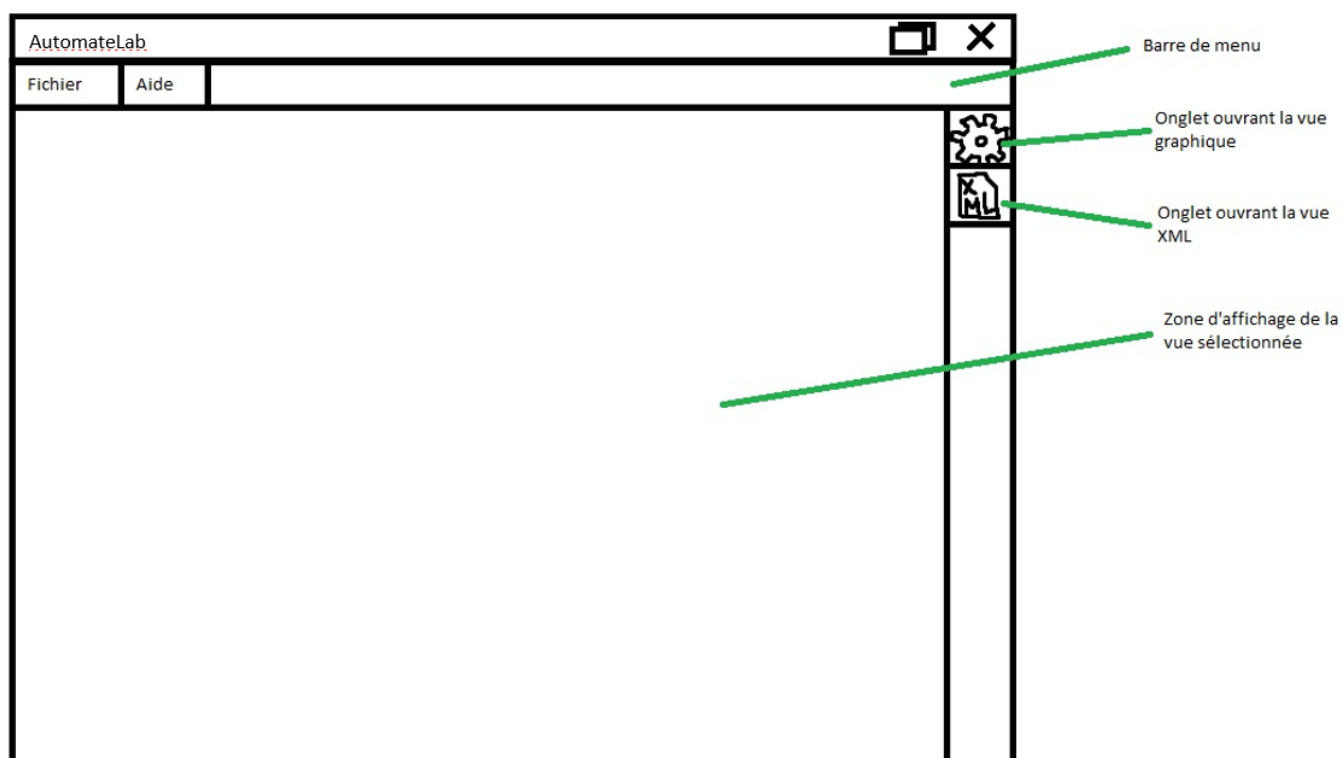


Fig. 1: IHM général de l'application

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

1.3. Hypothèses et risques

Ce projet comporte deux risques majeurs, liés aux biens sensibles du projet.

- Le premier est lié au projet en lui-même (en tant que bien: code source) et aux documents propres du projet (spécifications, tests, conception). Ce sont les biens à protéger. Les supports de ces biens sont nos machines, ainsi que les clouds utilisés pour synchroniser / stocker les données. Les risques sont donc le dysfonctionnement d'une ou plusieurs machines, d'un cloud ou d'un service de synchronisation. Il y a également le risque de faille de sécurité dans un de nos systèmes d'exploitation (Windows par exemple). En raison de la multiplicité des stockages (git, google drive, quatre machines), et de leur accessibilité, ce risque est finalement assez faible.
- Le bien le plus sensible est la note associée au projet. Elle n'a pas de bien support, la menace principale est le non-respect des délais.

1.4. Glossaire et abréviations

Abréviations

DTV : Dossier de Test Validation

PTV : Plan de Test Validation

T&V : Test et Validation

Glossaire

Campagne de Test :

Activité qui consiste à dérouler un ensemble de jeux de test. Un dossier de test est produit à l'issue d'une campagne.

Cas de Test :

Déclinaison d'un test précisant les valeurs utilisées pour les variables du test ainsi que les résultats attendus.

Dossier de Test :

Ensemble documentaire qui contient la description des scénarios et cas de tests, ainsi que l'exécution des jeux de test. Le dossier de test est le reflet d'une campagne de test.

Jeux de Test :

Ensemble de scénarios et de cas de tests permettant de tester un produit logiciel. L'enchaînement des cas et scénarios de tests est relatif à une stratégie de test précisée dans le plan de test.

Plan de Test (ou Protocole de Recette) :

Document décrivant le déroulement d'un jeu de test : Stratégie de test, critères d'arrêt, planification.

Scénarios de Tests :

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

Ensemble de cas de tests cohérents permettant de traiter un objectif fonctionnel.

Test Fonctionnel :

Test (vu de l'utilisateur) du bon fonctionnement d'un produit logiciel, d'une fonctionnalité ou d'une fonction de base. Vérification par rapport aux spécifications.

Test de Non Régression :

Vérification qu'une nouvelle version du produit logiciel fonctionne sans dégradation (technique, fonctionnelle, performance) par rapport à la version précédente.

Test Unitaire :

Vérification qu'un module, objet, ou autres composants d'un produit logiciel est conforme à ses spécifications (techniques et fonctionnelles). Basé sur des tests fonctionnels (vue de l'extérieur) et structurels (vue de l'intérieur).

Test ISO périmètre

Vérification que la nouvelle application développée réalise et couvre à l'identique l'ensemble des fonctionnalités présente dans l'ancienne application

Test de Validation :

Vérification que le produit logiciel est cohérent et complet par rapport aux spécifications fonctionnelles.

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

2. Organisation de la validation

2.1. Documents de tests

Les documents suivants sont générés par l'équipe de validation et constituent le référentiel documentaire de test.

Ces documents sont identifiés par :

- Dossier de Test de Validation.
- Dossier de Test Unitaire.

Les documents sont édités avec Google Doc.

Les données de test générées par les activités sont identifiées par :

- copie de tous les fichiers générés par les outils : scénarios, scripts de test, fichiers de données.
- sauvegarde de la base de données de référence (export de données).
- rapport de couverture lors de l'exécution des tests (avec ou sans outil).
- fiches d'anomalies.

Tous ces éléments doivent pouvoir être identifiés. Un rapport de validation regroupe et lie toutes ces informations.

2.2. Environnement de test

Nous utilisons les mêmes ordinateurs que lors de la programmation de l'application. Les environnements sont les suivants:

- Windows 10 version 21H2
- Windows 11 version 21H2
- macOS Ventura 13.1
- Linux Mint 21.1 Kernel version 5.15.0-57-generic

Java version 19.0.1 (minimum requis pour l'application)

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

3. Ressources

3.1. Ressources humaines

Les ressources nécessaires à la définition et à l'application des tests de validation sont listées dans le tableau suivant :

Equipe	Profil	Nombre
Équipe de développement	Développeur (pour anomalie)	4
Equipe de validation	Spécialiste outil tests	4

Ces profils sont utilisés suivant la planification des phases de tests et en fonction de l'importance du projet. Le tableau suivant indique pour chaque phase du projet les tâches allouées à chaque profil :

Tâche	Responsabilité	Effort (jour)	Observations
Préparer le Plan de Test	Equipe validation	2	
Spécification des Tests - Fonctions de gestion - Fonctions d'édition - fonctions interfaces - etc..	Equipe de validation	3	n écrans, n états, n traitements, n interfaces / jour sur la base de n écrans de niveau critique.

L'effort alloué à la résolution des problèmes rencontrés lors de l'exécution des scénarios comprend :

- l'analyse du problème par l'équipe de validation
- la détermination des modifications à apporter par le chef de projet.

3.2. Ressources matérielles

Les ressources matérielles nécessaires à l'accomplissement des tâches sont définies dans le tableau suivant :

Tâche	Matériel	Observations
Préparer le Plan de Test	Ressources prestataire	
Spécification des Tests	Outils bureautiques Spécifications fonctionnelles détaillées	Dernière version des documents

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

4. Stratégie de test

La stratégie de test de l'application repose sur un principe de couverture totale. Chaque use-case défini dans le dossier de spécifications sera couvert par au moins un test. Chaque test (unitaire, d'intégration ou de validation fonctionnelle) est caractérisé par sa fiche. Les fiches de tests unitaires sont spécifiées dans le document annexe "Dossier de tests unitaire.pdf", et les fiches de tests de validation dans le fichier annexe "Dossier de tests unitaire.pdf". Le projet ne contient pas de tests d'intégration.

4.1. Stratégie de validation

La stratégie de validation s'effectue en deux étapes :

- Tests unitaires
- Tests de validation fonctionnelle

Les IHM de l'application sont testés par les tests de validation fonctionnelle. L'application étant très orientée vers l'interface graphique, ce sont les principaux objets des tests.

4.1.1. Principe

L'objectif des tests de validation est d'assurer la conformité fonctionnelle de chaque module et de ses interfaces par rapport aux spécifications fonctionnelles et techniques. Il s'agit de parcourir l'ensemble des chemins du module en utilisant des jeux de valeurs valides et invalides. Pour cela on applique des séries de scénarios correspondant aux modèles des traitements.

Les scénarios doivent décliner les cas de test suivant la criticité de la fonctionnalité ou de la règle de gestion mise en œuvre.

4.1.2. Organisation de la stratégie

La stratégie de validation mise en place doit suivre les phases du développement (par lot), pour cette raison l'organisation est découpée en deux niveaux:

- 1er niveau: création des scénarios associés à chaque lot,
- 2ème niveau: validation globale des lots.

4.1.3. Objectif de test

Les objectifs de test associés à la validation sont identifiés comme suit:

Domaine de l'application

Les tests couvrent les objectifs fonctionnels: vérification de la conformité de traitement des fonctions. Ce domaine fonctionnel ne couvre que les fonctions critiques (voir chapitre suivant)

Les cas de test seront définis comme suit:

- Les fonctions définies comme critiques sont testées pour le cas nominal de traitement.
- Un cas d'erreur est appliqué aux règles de gestion mises en œuvre dans ces fonctions.

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

- Suivant la criticité des règles de gestion (définie par CP), plusieurs cas de test sont appliqués.

Domaine graphique

Il n'y a pas d'objectif graphique, c'est à dire que les objets graphiques sont testés dans la mesure où ils sont utilisés dans l'enchaînement des écrans :

Pas de contrôle systématique sur chaque objet graphique sauf lorsque le client l'impose.

Pas de contrôle du respect des règles d'ergonomie s'il n'existe pas de charte d'ergonomie transmise par le client.

4.1.3.1. Validation des règles de gestion

La stratégie de tests de conformité fonctionnelle appliquée sur l'application doit valider l'ensemble des règles de gestion définies pour l'application.

Pour chaque scénario, on prendra :

- des cas de tests correspondant au fonctionnement nominal (au moins 1 cas par classe d'équivalence),

Pour chacune de ces opérations les tests sur les saisies valides sont déclinés selon les cas de tests suivant :

- saisie avec les valeurs proposées dans les listes,
- saisie manuelle des valeurs dans les champs,
- saisie de valeurs moyennes.
- des cas de tests aux limites de la plage de validité (au moins 1 cas par limite),
Ces tests devront être effectués dans chacun des contextes d'utilisations possibles (profils utilisateurs).

4.1.3.2. Validation des règles de calcul

Toutes les règles de calcul seront testées avec des valeurs supérieures, inférieures et limites. L'ensemble des règles de calcul liées aux taux de TVA, aux montants multiples, aux sommes... doit être recensé et spécifié pour pouvoir être validé.

4.1.3.3. Validation des règles d'enchaînement

Le contrôle des règles d'enchaînement doit permettre le parcours de tous les enchaînements d'écrans dans chacun des cas d'utilisation possibles. Il sera nécessaire de tester les écrans d'édition afin de valider l'ensemble des états possibles de chacune des fenêtres de l'application. Ceci implique d'actionner tous les boutons possibles sur chaque écran.

4.1.3.4. Validation des procédures d'installation et d'exploitation

Concernant la validation des procédures d'installation et d'exploitation, aucune recette ne sera effectuée. A la livraison des lots correspondants, l'exécution des procédures permet de valider leur contenu.

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

4.1.4. Niveaux de criticité

Trois niveaux de criticité sont définis : **Critique**, **Courant** et **Annexe**. Dans le cadre des tests de validation l'affectation des niveaux de criticité aux fonctions est choisie selon la fréquence d'utilisation des fonctionnalités et leur impact sur la base de données.

Critique

Ce niveau correspond aux fonctionnalités mettant à jour la base de données (les fonctions les plus critiques étant celles qui effectuent des mises à jour irréversibles) ou aux fonctionnalités majeures de l'application (ex : routage de flux, interface, fonction dont le dysfonctionnement bloque l'application, etc..)

Courant

Ce niveau correspond aux fonctionnalités les plus utilisées en interrogation ou visualisation.

Annexe

Ce niveau correspond aux fonctionnalités n'ayant pas d'impact sur le traitement des données et aux fonctionnalités peu utilisées. Ce niveau regroupe toutes les éditions, certaines fonctions de consultation, des fonctions d'extraction... "

4.1.5. Critères d'arrêt des tests

4.1.5.1. Normal

Le **critère d'arrêt normal** des tests (au niveau de chaque campagne) est défini par le respect des conditions suivantes :

- tous les cas de tests ont été déroulés,
- tous les résultats obtenus sont analysés et enregistrés,
- toutes les fiches d'anomalies sont soldées.

4.1.5.2. Anormal

Le **critère d'arrêt anormal** d'une campagne intervient dans les cas suivants :

- ressources nécessaires à la campagne insuffisantes,
- déroulement non conforme,
- nombre d'anomalies détectées dans l'exécution est supérieur à 5.

Le premier cas est contrôlé lors de la préparation de l'exécution des tests (ex: base de données non à jour, réseau non disponible,...)

Les deux autres cas sont traités directement dans les scénarios de test par le contrôle et l'enregistrement des fichiers.

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

5. Description des tests

5.1. Validation des projets

5.1.1. Dossier de tests unitaire

Le dossier de tests unitaires est initialisé dans la phase de conception détaillée et finalisé avant la phase de tests unitaires. Il contient l'ensemble des tests par module. En fonction de la complexité de l'application, on peut établir un dossier par lot fonctionnel. L'objectif est de valider la qualité du code et les performances de chacun des modules développés.

Le dossier est en annexe: "Dossier de tests unitaire".

5.1.1.1. Correspondance

Les correspondances entre numérotation et objets vérifiés par les fiches de tests unitaires sont explicitées en [Figure 3](#).

5.1.2. Dossier de tests de validation

Le dossier de tests de validation est initialisé dans les phases de spécification et finalisé avant la phase de validation. On effectuera dans un premier temps les tests de stabilité pour vérifier les performances globales de l'application. Dans un second, on effectuera les tests fonctionnels correspondant aux besoins du client.

Le dossier est en annexe: "Dossier de tests de validation".

5.1.2.1. Simulation des utilisateurs

Pour ce projet, la simulation est réalisée par des utilisateurs de l'application.

5.1.2.2. Correspondance

Les correspondances entre numérotation et objets vérifiés par les fiches de tests unitaires sont explicitées en [Figure 3](#).

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

5.2. Matrice de couverture

On donne la matrice de couverture des fonctionnalités par les fiches de tests à la page suivante ([Fig. 4](#)).

Par commodité de lecture, on donne les tableaux intitulés/libellés des use-cases ([Fig. 2](#)) et fiches de tests ([Fig. 3](#)) ci-dessous:

PRJ-i	PRJ-i-j	Libellé
PRJ-001		Ouverture de l'application
PRJ-002		Ouvrir un fichier
	PRJ-002-1	Charger le fichier par défaut
	PRJ-002-2	Charger un automate depuis un fichier XML
PRJ-003		Visualiser un automate (vue graphique et XML)
PRJ-004		Editer un automate (vue graphique et XML)
	PRJ-004-1	Vue graphique - Création d'un état
	PRJ-004-2	Vue graphique - Suppression d'un état
	PRJ-004-3	Vue graphique - Création d'une transition
	PRJ-004-4	Vue graphique - Suppression d'une transition
	PRJ-004-5	Vue graphique - Modification d'une transition
	PRJ-004-6	Vue graphique - Attribution/désattribution de la propriété état initial/final à un état
PRJ-005		Sauvegarder un automate en fichier XML
	PRJ-005-1	Enregistrer
	PRJ-005-2	Enregistrer sous
PRJ-006		Fermeture de l'application

Fig. 2: Correspondances entre numérotation et libellé des use-cases

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

FT-i	Objet(s) vérifié(s) par le test
FT-01	L'application se lance, et affiche un écran vide
FT-02	Création d'un nouvel automate (chargement du fichier par défaut)
FT-03	Chargement d'un automate depuis un fichier XML
FT-04	Affichage de l'automate dans les vues disponibles (graphique, XML)
FT-05	Edition dans la boîte d'entrée texte de la vue XML
FT-06	Création d'un état (vue graphique)
FT-07	Suppression d'un état (vue graphique)
FT-08	Création d'une transition (vue graphique)
FT-09	Suppression d'une transition (vue graphique)
FT-10	Modification d'une transition (vue graphique)
FT-11	Attribution/désattribution de la propriété "isInitial" à un état (vue graphique)
FT-12	Attribution/désattribution de la propriété "isFinal" à un état (vue graphique)
FT-13	Enregistrement d'un fichier sur disque, ayant déjà été enregistré sur disque
FT-14	Enregistrement d'un fichier sur disque
FT-15	Fermeture de l'application
FTU-i	Objet(s) vérifié(s) par le test
FTU-01	Fonction loadFile()
FTU-02	Fonction displayPopUp()
FTU-03	Fonction parseXML()
FTU-04	Fonction loadDefaultFile()
FTU-05	Fonction updateModel()
FTU-06	Fonction pullModel()
FTU-07	Fonction createState()
FTU-08	Fonction getNextFreeEtatNumber()

Ref : ST-8-0A Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
---	-------------------------------	---

FTU-09	Fonction deleteState()
FTU-10	Fonction createTransition()
FTU-11	Fonction deleteTransition()
FTU-12	Fonction editTransition()
FTU-13	Fonction setStateInitial()
FTU-14	Fonction setStateFinal()
FTU-15	Fonction toXML()
FTU-16	Fonction saveCurrentFile()

Fig. 3: Correspondances entre numérotation et objets vérifiés par les fiches de tests

Ref : ST-8-0A	Projet Editeur d'automates	Date: 08/01/2023 Version : 0B Service : Polytech Marseille Etat : Préliminaire
Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates		

PRJ-i	PRJ-i-j	Fiche de tests FT-i															Fiche de tests unitaire FT-j															
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
PRJ-001																																
PRJ-002																																
	PRJ-002-1																															
	PRJ-002-2																															
PRJ-003																																
PRJ-004																																
	PRJ-004-1																															
	PRJ-004-2																															
	PRJ-004-3																															
	PRJ-004-4																															
	PRJ-004-5																															
	PRJ-004-6																															
PRJ-005																																
	PRJ-005-1																															
	PRJ-005-2																															
PRJ-006																																

Fig. 4: Matrice de couverture des use-cases par les fiches de tests

	Le test en abscisse n'est pas nécessaire au bon fonctionnement du use-case en ordonnée
	Le test est nécessaire au fonctionnement du use-case
	Le test est nécessaire au fonctionnement du use-case, par dépendance d'un autre use-case (<<include>>)