

Cu N° Révision	Date	Auteur(s)	Commentaires
0.0	27/03/2016	Hubert GOUILLARD	Création d'une première version du dossier de spécification.
0.1	28/03/2016	Hubert GOUILLARD	Formalisation du document selon modèle défini par le responsable qualité.
0.2	28/03/2016	Laura GESLIN	Relecture, modification des cas d'utilisation.
0.3	29/03/2016	Hubert GOUILLARD	Remise en forme, agencement des parties. Ajout des parties manquantes.
0.4	29/03/2016	Sébastien BOTTE	Relecture du dossier. Corrections d'erreurs.
0.5	30/03/2016	Clarisse GIRAULT	Modification des cas d'utilisation, vérification des architectures et description des fonctions.
0.6	30/03/2016	Hubert GOUILLARD	Relecture du dossier. Mise à jour du document selon modifications décidées en réunion interne.
1.0	8/04/2016	Hubert GOUILLARD	Ajout de la variante générale du projet dans le cas d'ajout de la carte Thingy_Elektronics. Modification de la partie IHM selon conseils audit. Modification architecture et contexte selon réunion client
1.1	20/04/2016	Sébastien BOTTE	Correction de la numérotation de l'arborescence.
1.2	21/04/2016	Laura GESLIN	Modification des contextes suite au développement du dossier de conception générale
1.3	22/04/2016	Guillaume MURET	Modification apportée par la nouvelle machine à état de l'IHM
1.4	01/05/2016	Hubert GOUILLARD	Relecture et corrections mineures. Remise au format client.
1.5	01/05/2016	Laura GESLIN	Relecture, corrections et validation
2.0	11/05/2016	Guillaume MURET	Modification : plus de date pour « last connection »
2.0	11/05/2016	Laura GESLIN	Modification du CU Identifier l'opérateur : ajout d'un accès au serveur de test
2.0	12/05/2016	Laura GESLIN	Modification CU Lancer les tests suite bug enregistrement (conception)

2.1	13/05/2016	Laura GESLIN	Ajout terme « anciens tests » dans dictionnaire du domaine, modification contexte physique suite Bug test de communication (conception) et mise en relecture
2.2	15/05/2016	Guillaume MURET	Modif du cu lancer tests (modification de l'ordre du déroulement des tests)
2.2	15/05/2016	Guillaume MURET	Relecture
2.2	15/05/2016	Laura GESLIN	Validation du document pour AN 18/05/2016
2.3	18/05/2016	Laura GESLIN	Ajout détail pour le protocole de communication, remise en relecture
2.3	18/05/2016	Hubert GOUILLARD	Relecture complète du dossier.
2.3	18/05/16	Guillaume MURET	Relecture
3.0	19/05/16	Laura GESLIN	Validation

Thingy – GCE Electronics

Dossier de spécification

COPYRIGHT :

Le présent document est la propriété de PRØVE. Il est diffusé pour les seuls besoins du projet concerné. Il ne doit pas être reproduit, entièrement ou partiellement, ou employé pour tout autre but. Aucune information quant au contenu ou aux thèmes de ce document ne peut être communiquée de quelque façon à un tiers sans autorisation écrite de PRØVE. L'équipe de consultants (société FORMATO) du projet ProSE échappe à ces restrictions et peut utiliser le présent document pour toute utilisation qui leur convient, et ce sans autorisation de PRØVE. Le client de part le contrat signé, est autorisé à utiliser le document pour les besoins du projet mais ne possède pas les droits de modification ou de diffusion du présent document.

Table des matières

I. Introduction.....	6
1.1 Objet.....	6
1.2 Portée.....	7
1.3 Définitions, acronymes et abréviations.....	8
1.4 Références.....	10
1.5 Vue d'ensemble.....	11
2. Description générale.....	12
2.1 Caractéristiques des acteurs.....	13
2.1.1 Acteurs directs.....	13
2.2 Environnement.....	14
2.2.1 Architecture matérielle et logicielle.....	14
2.2.2 Les interfaces du système.....	19
2.2.2.1 Les Interfaces logiques.....	19
2.2.2.2 Les interfaces avec les acteurs.....	21
2.2.2.2.1 En provenance de l'opérateur.....	21
2.2.2.2.2 A destination de l'opérateur.....	22
2.2.2.2.3 En provenance de l'IPX800V4.....	23
2.2.2.2.4 A destination de l'IPX800V4.....	24
2.2.2.2.5 En provenance du serveur de tests.....	25
2.2.2.2.6 A destination du serveur de tests.....	25
2.2.2.2b Les interfaces avec les acteurs.....	27
2.2.2.2b.1 A destination de l'IPX800V4.....	27
2.2.2.3 Les interfaces physiques.....	29
2.2.2.3.1.a E_X8R.....	29
2.2.2.3.1.b E_X8R.....	31
2.2.2.5 Les interfaces de communication.....	31
2.2.3 Les contraintes de mémoire.....	35
2.2.4 Contraintes matérielles.....	35
2.2.5 Exigences d'adaptation.....	35
2.3 Fonctions principales développées.....	36
2.3.1 Rappel sur les cas d'usage.....	36
2.3.2 Rappel sur les cas d'utilisation.....	36
2.3.2.1 Représentation graphique des CUs.....	37
2.3.2.2 Représentation textuelle des CUs.....	38
2.3.3 Résumé des cas d'usage considérés pour Prøve.....	38
2.3.4 Résumé des cas d'utilisations stratégiques.....	39
2.3.5 CU_01 Démarrer Thingy.....	41
2.3.5.1 Description textuelle.....	41

2.3.6 CU_02 Identifier l'opérateur.....	42
2.3.6.1 Description textuelle.....	42
2.3.7 CU_03 Lancer les tests.....	43
2.3.7.1 Description graphique.....	43
2.3.7.2 Description textuelle.....	43
2.3.8 CU_04 Identifier la carte.....	46
2.3.8.1 Description textuelle.....	46
2.3.9 CU_05 Configurer les tests.....	47
2.3.9.1 Description textuelle.....	47
2.3.10 CU_06 Consulter les rapports des anciens tests.....	48
2.3.10.1 Description textuelle.....	48
2.3.11 CU_07 Consulter les statistiques.....	49
2.3.11.1 Description textuelle.....	49
2.3.12 CU_09 Déconnecter Opérateur.....	50
2.3.12.1 Description textuelle.....	50
2.3.13 CU_010 Quitter Thingy.....	51
2.3.13.1 Description textuelle.....	51
3. Exigences spécifiques.....	52
3.1 Interface Homme-Machine.....	52
3.1.2 Vue générale.....	52
3.1.3 Les actions utilisateur.....	59
3.1.4 Les écrans et dialogues.....	64
3.2 Description des fonctions.....	97
3.2.1 Vérifier les identifiants de l'opérateur.....	97
3.2.2 Vérifier le choix des tests à réaliser.....	97
3.2.3 Réaliser une sélection de tests.....	98
3.2.4 Consulter l'historique.....	98
3.2.5 Visualisation des statistiques des tests.....	99
.....	99
3.2.6 Interrompre pour cause d'erreur de communication.....	99
3.2.7 Vérifier la validation du test.....	100
3.2.8 Confirmer le choix de quitter l'écran de résultats.....	100
3.2.9 Enregistrer les résultats des tests.....	101
3.3 Dictionnaire du domaine.....	102

I. Introduction

1.1 Objet

Ce dossier de spécification définit les fonctionnalités et exigences attendues par la société cliente GCE Electronics pour le développement logiciel d'un banc de test domotique. Ce banc de test s'appuie sur le produit IPX800V4 – Automate Webserver fourni par la société cliente, et repose sur l'utilisation d'un nano-ordinateur Raspberry. Il devra cependant être maintenable et adaptable afin de faciliter l'intégration pour la gestion potentielle d'autres produits. Ce développement prendra en compte les différents cas d'utilisation et visera principalement à réduire le temps d'exécution des tests actuels. La conception favorisera un affichage des résultats simple et intuitif.

Ce dossier fixera les objectifs de conception et de réalisation. Il permettra à l'équipe de test de concevoir et réaliser des tests couvrant l'intégralité des objectifs fixés par le cahier des charges.

Les principaux objectifs sont donc l'automatisation et l'optimisation des tests physiques de la carte, des tests de communication de la carte et des tests de viabilité hardware et firmware. Ces objectifs ont été déterminés à la suite de l'étude du cahier des charges [CDCProSe_GCEv2_SANS_EA2], d'une étude du dossier API [API-IPX800V4], et d'une rencontre avec M.Labbé, représentant de l'entreprise GCE Electronics et M.Delatour de l'entreprise FORMATO (voir compte-rendu et ordre du jour de la réunion [CR 04-03-2016_CR-C_EA2] [ODJ 04-03-2016_OJ-C_EA2]).

Ce dossier de spécification respecte les exigences pour les dossiers de spécification sur les logiciels 830-1998 – IEEE. De plus, il suit les exigences du Plan d'Assurance Qualité Logicielle (PAQL) [PAQL_EquipeA2] défini par la société PRØVE sous la direction de la société FORMATO.

1.2 Portée

Dans ce contexte, le Système à l'Étude (SaE) est constitué :

- Du logiciel Thingy_Commande (TC), permettant de réaliser les tests des entrées/sorties de l'IPX800V4 - Automate Webserver ; ainsi que la gestion de l'identification des produits et des opérateurs,
- Du logiciel Thingy_App (TA), fournissant une interface ergonomique sur tablette qui permettra le choix des tests à réaliser et la visualisation des résultats obtenus,
- La carte Thingy_Hard (TH) sur laquelle s'exécutera TC.

Le Système à l'Étude servira à contrôler un certain nombre de périphériques (relais, capteurs, compteurs entre autres) voués à être testés. Les résultats obtenus ne seront pas traités directement par l'application. Des caractéristiques et normes de traitement devront être respectées afin de faciliter la compréhension des résultats et son traitement ultérieur. Le SaE devra également prendre en compte les contraintes imposées par le produit support IPX800V4.

Il sera de la responsabilité du client de :

- Fournir l'objet d'étude IPX800V4 sur lequel s'appuiera les tests,
- Fournir l'application permettant la visualisation des relevés capteurs,
- Procéder à l'intégration du logiciel TA sur des tablettes dédiées aux tests.

Cette spécification ne traitera pas des périphériques de mesure, ni des caractéristiques du produit support IPX800V4. Il ne détaillera pas les exigences de ce produit support.

1.3 Définitions, acronymes et abréviations

Les acronymes et définitions utilisés dans ce document seront répertoriés dans le tableau joint ci-dessous. Le vocabulaire technique et spécifique au domaine sera défini en fin de dossier dans le dictionnaire de domaine se trouvant dans la partie 3.3 Dictionnaire du domaine.

Acronymes, abréviations	Définitions
Client	Société GCE Electronics, N° SIRET : 53181968800023 Représentée par M. Labbé.
CU	Cas d'utilisation.
Différentiel TTL	Le différentiel TTL est un type de signal électrique binaire respectant les normes Transistor-Transistor Logic.
Disponibilité	La disponibilité est l'aptitude d'un composant ou d'un système à être en état de marche à un instant donné.
EnOcean	Capteurs radio utilisant une technique de transmission radio sans piles grâce à des interrupteurs piézo-électriques.
Fiabilité	La fiabilité est l'aptitude d'un composant ou d'un système à fonctionner pendant un intervalle de temps.
IEEE (Institute of Electrical and Electronics Engineers)	Association professionnelle inter-nationale définissant entre autres des normes dans le domaine informatique et électronique.
IHM (Interface Homme Machine)	Moyens permettant aux utilisateurs de Thingy d'interagir avec TC.
Maintenabilité	La maintenabilité est l'aptitude d'un composant ou d'un système à être maintenu ou remis en état de fonctionnement.
OMG (Object Management Group)	Association professionnelle internationale définissant entre autres des normes dans le domaine informatique.
SaE (Système à l'étude)	Il s'agit de l'ensemble des composants TA et TC.
Sûreté de	La sûreté de fonctionnement est l'aptitude d'une entité

Fonctionnement (SdF)	à satisfaire une ou plusieurs fonctions requises dans des conditions données.
SVN (Thingy)	Dépôt numérique des documents du projet Bænk. Ce dépôt est mis à la disposition de l'équipe de développement FORMATO. Pour avoir accès à ce dépôt, merci de prendre contact avec M. Delatour de la société FORMATO ou à défaut, de toute personne en charge du projet Bænk au sein de FORMATO.
TTL (<i>Transistor-Transistor Logic</i>)	Le Transistor-Transistor Logic est une famille de circuits logiques utilisés en électronique. Il définit une norme binaire électrique possédant un état « bas » compris entre 0V et 0,8V, et un état « haut » compris entre 2V et 5V.
UART (<i>Universal Asynchronous Receiver Transmitter</i>)	Composant utilisé pour les liaisons entre un ordinateur et un port série par envoi de données en parallèle.
UML (<i>Unified Modeling Language</i>)	Notation graphique normalisée, définie par l'OMG et utilisée en génie logiciel.
USB (<i>Universal Serial Bus</i>)	Norme décrivant un bus informatique en transmission série qui sert à connecter des périphériques informatiques à un système informatique.

1.4 Références

Le tableau ci-dessous répertorie les documents cités dans ce dossier de spécification ainsi que les liens permettant d'y accéder.

Nom du document	Références du document
[API-IPX800V4]	Application Programming Interface : interface de programmation de la carte IPX800V4 fournie par la société GCE Electronics présente sur le SVN de l'équipe PRØVE
CDCProSe_GCEv1_SANS_EA2	Société GCE Electronics, « Cahier des charges pour le développement d'un banc de test domotique pour IPX800 V4 - Automate Webserver », version 1.0, Référentiel Documentaire Projet Bænk, 2016
CDCProSe_GCEv2_SANS_EA2	Société GCE Electronics, « Cahier des charges pour le développement d'un banc de test domotique pour IPX800 V4 - Automate Webserver », version 2.0, Référentiel Documentaire Projet Bænk, 2016
CR 04-03-2016_CR-C_EA2	Prøve, « compte rendus de réunion sur la spécification Thingy avec M. Labbé en date du 04/03/2016 », Référentiel Documentaire Projet Bænk, 2016.
CR_17-03-2016_CR-C_EA2	Prøve, « compte rendus de réunion sur la spécification Thingy avec M. Labbé en date du 17/03/2016 », Référentiel Documentaire Projet Bænk, 2016.
[GNU-GPL_3.0_2007]	GNU General Public Licence, version 3.0, http://www.gnu.org/licenses/gpl-3.0.html , 2007.
ME_IPX800_V4	Société GCE Electronics « Mode d'emploi IPX800 V4 » Référentiel Documentaire Projet Bænk. 2016.
ME_X8-R	Société GCE Electronics « Mode d'emploi X8-R »

	Référentiel Documentaire Projet Bænk. 2016.
[IEEE-830_1998]	IEEE, std 830-1998 « Recommended Practice for Software Requirements Specifications », http://standards.ieee.org/findstds/standard/830-1998.html , 1998.
PAQL_EquipeA2	Prøve, « Plan d'Assurance Qualité Logicielle », Référentiel Documentaire Projet Bænk, 2016

1.5 Vue d'ensemble

Ce document de spécification se structure en trois parties :

- La première partie « *Introduction* » présente les objectifs et la portée de la spécification,
- La deuxième partie « *Description générale* » présente l'environnement, le contexte de développement et les fonctionnalités principales,
- La troisième partie « *Exigences spécifiques* » présente de manière détaillée les IHM, les fonctionnalités particulières et le dictionnaire du domaine.

2. Description générale

La centrale domotique IPX800V4 est un dispositif de gestion de périphériques domotiques (chauffage, arrosage, volet roulant, ...) à usage domestique ou industriel. Celle-ci a l'apparence d'un compteur électrique.

Le banc de test domotique Thingy a pour but d'optimiser et d'effectuer le suivi des tests physiques, de communication et de viabilité (hardware et firmware) concernant la centrale domotique IPX800V4. Les relais nécessaires à la vérification sont pilotés à l'aide d'une carte Raspberry. L'opérateur effectuant ces tests y parvient via une application Android installée sur une tablette Android dédiée.

Ce banc de test s'inscrit d'une part dans une optique d'automatisation de ces tests de façon à pouvoir gagner en rapidité d'exécution. D'autre part, il y a une volonté de conservation d'une trace des tests effectués. Ceux-ci doivent apparaître clairement pour un utilisateur non-initié. De plus, ces données pourront être stockées dans un prototype de base de données utilisateur référencé grâce à un identifiant spécifique à chaque centrale.

La connexion au réseau Ethernet fait elle aussi partie des paramètres indispensables à vérifier pour l'enregistrement des données sur le prototype de base de données hébergé sur le serveur web de la carte Raspberry (qui sera ensuite déplacé vers le serveur propre à GCE).

La IPX800V4 est alimentée par une tension continue de 12V. Elle comporte par ailleurs diverses fonctions citées ci-dessous :

- 8 entrées digitales tout ou rien (contact sec). Extensible à 56.
- 8 sorties relais sur contact 270 V/10A extensible à 56.
- 4 entrées analogiques 10 bits.
- 32 Ping Watchdog (détection de périphériques réseaux par PING ICMP).
- 32 Périphériques EnOcean. (Nécessite l'extension X-ENO).
- Notifications GET et POST sécurisées: Cryptage des pushes en TLS (HTTPS).
- Support du Dyn-Dns.
- Moteurs de scénarios GX© : logique combinatoire ET / OU / NOT.
- Éditeur JavaScript pour personnaliser et créer ses propres widgets.
- Sources de données configurable JSON et XML.

2.1 Caractéristiques des acteurs

Par le terme d'acteur, nous désignons tout rôle joué par une entité (morale ou physique) qui interagit directement ou non avec le SaE, Thingy. Cette entité peut être une personne (généralement un utilisateur du système) ou un autre système.

Nous distinguons les acteurs, dits directs (qui interagissent directement avec Thingy) et les acteurs dits hors champs (qui n'ont pas d'interaction directe avec Thingy) mais qui sont à l'origine d'exigences à respecter par le SaE.

2.1.1 Acteurs directs

Les acteurs directs sont :

- **l'Opérateur** : utilisateur principal de Thingy. Le banc de tests, selon la commande sélectionnée via l'application, lui renvoie les résultats obtenus.
- **l'IPX800V4** : la centrale domotique que permet de tester le système à l'étude, Thingy.
- **le ServeurDeTest** : le prototype d'une base de données référençant les données nécessaires au fonctionnement de l'application.

2.2 Environnement

2.2.1 Architecture matérielle et logicielle

Thingy_Hard se compose d'une carte Raspberry fonctionnant sur un système d'exploitation de famille GNU/Linux. Dans le cas de la deuxième architecture proposée, TH disposera de plus, d'une carte électronique nommée Thingy_Elektronisk (dont l'intérêt de l'utilisation sera expliqué au chapitre 2.2.2.1 Les Interfaces logiques).

Thingy_Soft, quant à lui, se compose des entités Thingy_Commande et Thingy_App. (cf 3.3 Dictionnaire du domaine).

Nous ne développerons pas ici les aspects de maintenance et fourniture d'énergie.

Les conventions graphiques utilisées sont explicitées en figure Illustration 1. Ces diagrammes de déploiement identifient les entités matérielles et/ou logicielles avec lesquelles le SaE (composé des entités Thingy_Hard et Thingy_Soft) doit interagir et permet ainsi de déterminer les principaux échanges qu'il entretient avec son environnement.

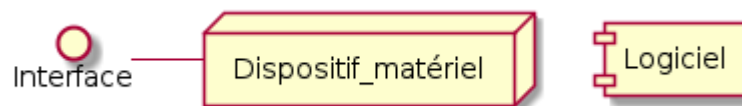


Illustration 1: Légende du diagramme de déploiement UML

Deux architectures seront présentées dans ce paragraphe. En effet, l'une présentera la solution initiale proposée par le client tandis que l'autre intégrera au système une carte électronique conçue dans le but de préciser la réalisation des tests sur la carte IPX800V4.

En fonction du temps nécessaire à la conception et aux tests fonctionnels de la carte électronique, l'une ou l'autre des ces deux architectures sera choisie pour la concrétisation du projet.

Première architecture envisagée

Architecture matérielle et logicielle imposée

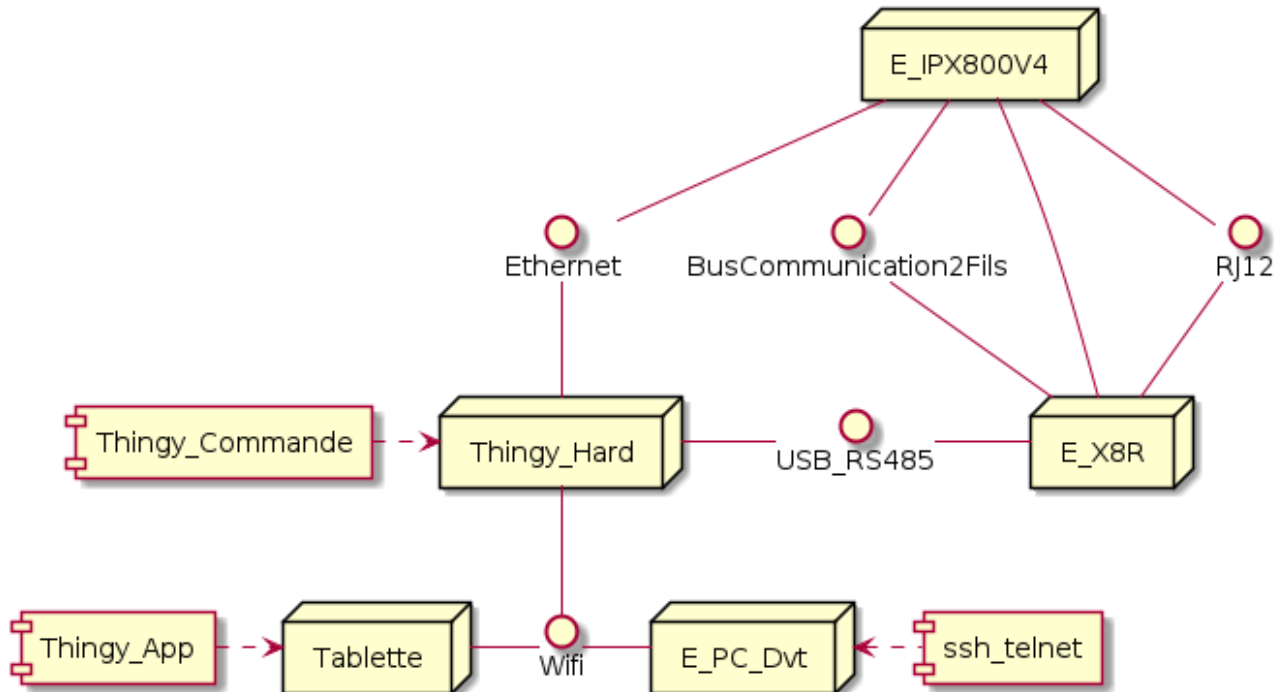


Illustration 2: Architecture matérielle et logicielle imposée sans TE

Comme indiqué sur l'illustration 2, Thingy Hard (représenté au centre) est en interaction avec différentes entités externes au SaE. Par convention, le nom de ces entités est préfixé par les lettres «E_» (E pour Externe), elles sont aussi désignées, dans ce document, par le terme de «périphérique Thingy».

En fonction des ordres donnés par l'opérateur, Thingy_Soft commandera ou non, suivant différentes politiques de lancement, l'automatisation des tests de la carte IPX800V4 à disposition via Thingy_Hard et par le biais de l'utilisation d'une carte X8R, considérée comme une extension de l'IPX800V4. Ces ordres de l'opérateur sont envoyés par l'intermédiaire de l'IHM, à savoir une tablette 7" Clip Sonic Android version 4.2 dédiée (Tablette).

Par ailleurs le SaE sera alimenté en utilisant une source d'énergie électrique via une connexion micro-USB.

La liste des périphériques Thingy est donc la suivante :

- Tablette : Il s'agit d'une tablette Android de version 4.2 minima, jouant le rôle d'IHM avec l'opérateur.
- E_PC_Dvt : ordinateur de développement permettant de faciliter le lancement de l'application sur Thingy_Hard ainsi que le débogage lors de la réalisation de l'application Thingy_Soft.
- E_IPX800V4 : carte destinée à être testée.
- E_X8R : extension modifiée de l'IPX800V4 permettant la réalisation des tests sur celle-ci.

Pour communiquer avec ces différents périphériques, Thingy devra, pour certains périphériques, utiliser et respecter des protocoles de communication spécifiques. Ces protocoles (désignés par le terme d'interface en UML) ainsi que le protocole de communication avec l'acteur principal dénommé IPX800V4 seront aussi présentés sur l'Illustration 4 (page 20).

Ainsi, la communication avec Tablette se fera par Wi-fi.

Le périphérique E_PC_Dvt devra être connecté par une liaison Wi-fi à Thingy_Hard, et plus précisément à Thingy_Rasp.

Enfin, Thingy_Hard et l'IPX800V4 seront reliés par une liaison Ethernet.

La liaison entre E_IPX800V4 et son extension E_X8R se fera par un bus de communication 2 fils, un câble RJ12 et une liaison filaire simple.

Ces protocoles et interfaces sont décrits plus précisément au chapitre 2.2.2.2 Les interfaces avec les acteurs.

Seconde architecture envisagée

Architecture matérielle et logicielle imposée

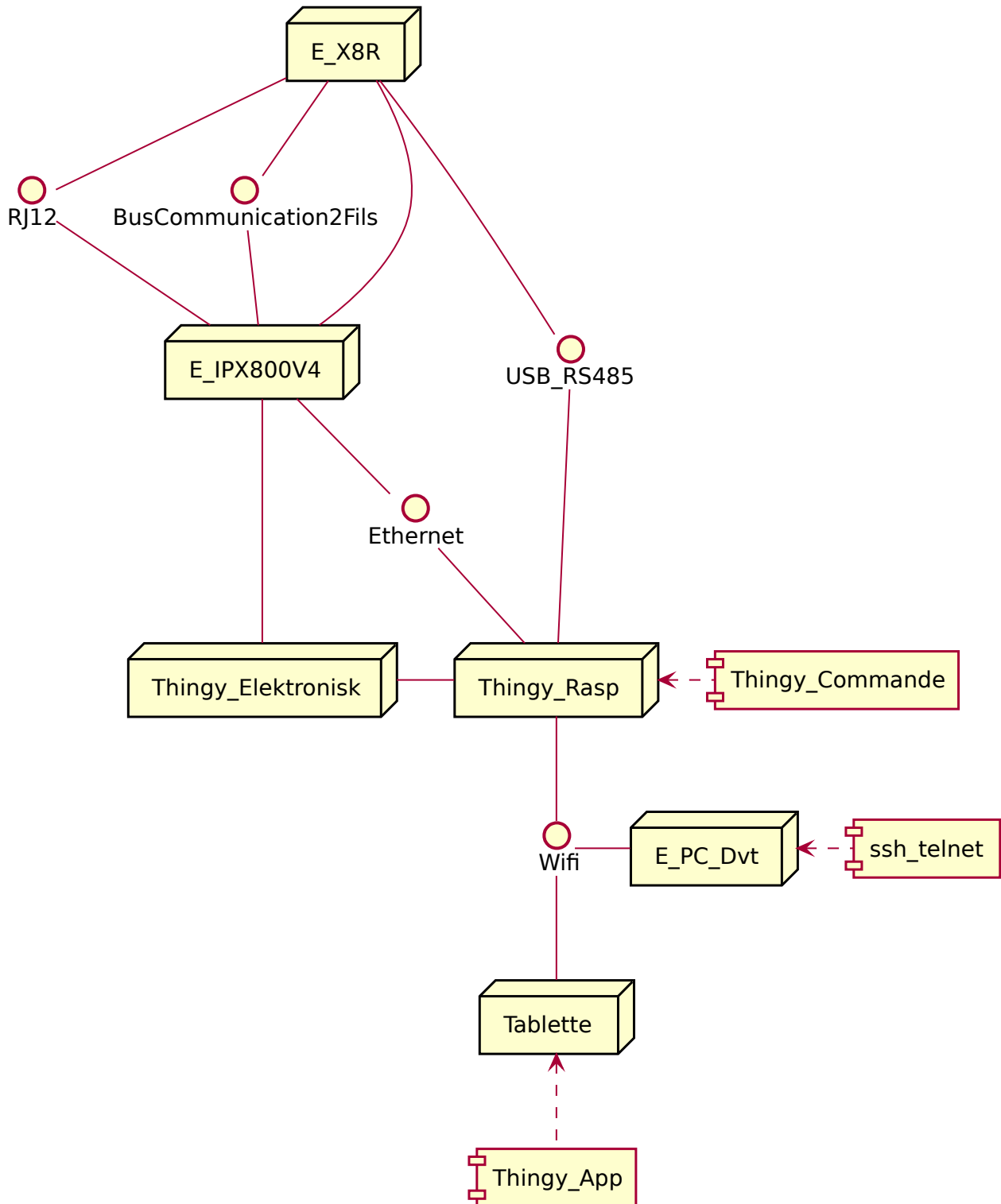


Illustration 3: Architecture matérielle et logicielle imposée avec TE

Les périphériques de cette architecture sont les mêmes que ceux présentés au paragraphe précédent (cf Première architecture envisagée).

Comme indiqué sur l'Illustration 3, Thingy Hard (représenté au centre par les deux entités Thingy_Elektronisk et Thingy_Rasp) est en interaction avec différentes entités externes au SaE.

Les protocoles de communication utilisés par le système ainsi présenté sont les mêmes que ceux explicités au paragraphe précédent (cf Première architecture envisagée), excepté les protocoles liant Thingy_Hard et l'IPX800V4.

En effet, ces deux entités seront, pour cette architecture, reliés par :

- une liaison Ethernet pour la liaison avec Thingy_Rasp,
- une liaison filaire pour la liaison avec Thingy_Elektronisk.

Ces protocoles et interfaces sont décrits plus précisément au chapitre 2.2.2.5 Les interfaces de communication

2.2.2 Les interfaces du système

Ce chapitre décrit les entrées et sorties dites « logiques » et « physique » du SaE. En effet, nous différencions dans cette étude deux grands types d'entrées/sorties :

- Celles dites de haut niveau (dites aussi logiques) qui décrivent les événements et données échangés entre l'utilisateur et les périphériques Thingy, ces entrées/sorties étant traitées ou pré-traitées par les périphériques Thingy avant d'être transmis au SaE. Ces entrées et sorties portent sur les intentions des acteurs interagissant avec le SAE.
- Celles dites de bas-niveau (dites aussi physiques) qui sont les entrées/sorties réellement échangées entre le SaE et les périphériques Thingy. Les entrées/sorties physiques (ou bas niveau) seront décrites au chapitre 2.2.2.3 Les interfaces physiques page 29.

2.2.2.1 Les Interfaces logiques

Deux interfaces logiques seront présentées dans ce paragraphe. En effet, l'une présentera la solution initiale proposée par le client tandis que l'autre intégrera au système une carte électronique TE conçue dans le but de préciser la réalisation des tests sur la carte IPX800V4.

En fonction du temps nécessaire à la conception et aux tests fonctionnels de la carte électronique, l'une ou l'autre des ces deux interfaces logiques sera choisie pour la concrétisation du projet.

Première interface logique envisagée : pas d'utilisation de TE

L'Illustration 4 (page 20) présente le contexte de Thingy en faisant figurer les entrées/sorties dites de haut niveau (ou logiques). Elles sont regroupées en grande famille. Pour représenter ce contexte logique, un diagramme de communication UML a été utilisé. Nous retrouvons les périphériques Thingy déjà présentés en Illustration 2 (page 15).

Contexte logique du SaE

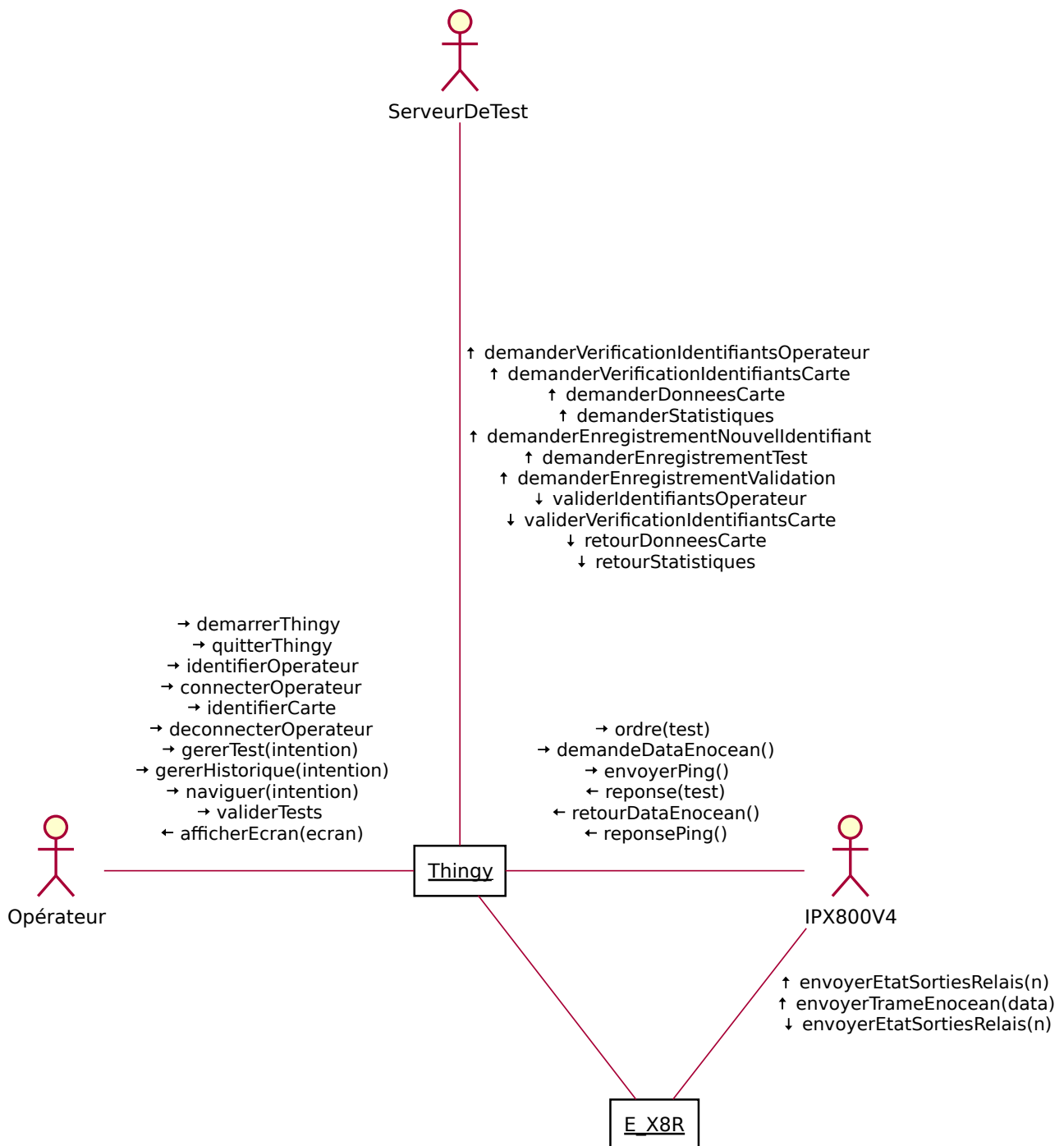


Illustration 4: Contexte logique du SaE sans TE

Dans ce diagramme de communication, seules les entrées/sorties logiques entre les acteurs et les périphériques Thingy sont présentées.

2.2.2.2 Les interfaces avec les acteurs

Les interactions entre les acteurs (opérateur, maintenance et fournisseur d'énergie) de Thingy se font par l'intermédiaire des périphériques suivants :

- Tablette
- E_PC_Dvt
- E_X8R

Nous allons maintenant détailler ces entrées et sorties logiques.

2.2.2.2.1 En provenance de l'opérateur

2.2.2.2.1.1 Vers Thingy

Voici les événements logiques en provenance de l'opérateur vers Tablette :

- demarrerThingy : signale que l'opérateur démarre Thingy
- quitterThingy : signale que l'opérateur éteint Thingy.
- IdentifierOperateur : signale que l'opérateur s'identifie (indication de son nom et de son mot de passe).
- IdentifierCarte : signale que l'opérateur identifie une carte IPX800V4 qu'il veut tester (indication du numéro de la carte ou génération automatique de l'identifiant).
- connecterOperateur : signale que l'opérateur se connecte
- identifierCarte : signale que l'opérateur identifie une carte IPX800V4 qu'il veut tester (indication du numéro de la carte ou génération automatique de l'identifiant).
- deconnecterOperateur : signale que l'opérateur se déconnecte.
- gererTest(intention) : signale que l'opérateur souhaite accéder à la gestion d'un test. Les intentions correspondantes à ces actions sont les suivantes :
 - lancer un nouveau test
 - configurer un test
 - démarrer un test
 - rejouer un test
- gererHistorique(intention) : signale que l'opérateur souhaite accéder à des données présentes dans le serveur de test. Les différentes intentions correspondant à cette action sont les suivantes :
 - consulter l'historique des tests relatifs à une carte
 - consulter l'historique des derniers tests effectués
 - identifier la carte à rechercher pour en consulter l'historique
 - consulter les statistiques des tests
- naviguer(intention) : signale que l'opérateur souhaite valider, annuler ou quitter l'action en cours. Les intentions correspondant à ces actions sont les suivantes :

- valider
- annuler
- quitter
- reconnecter
- validerTests : signale que l'opérateur souhaite valider ou invalider la suite de tests venant d'être réalisée sur une carte IPX800V4

2.2.2.2.2 A destination de l'opérateur

2.2.2.2.2.1 Depuis Thingy

Voici les événements logiques de Tablette vers l'opérateur:

- afficherEcran(ecran) : la tablette affiche les différents menus en réponse aux demandes de l'opérateur. Les différents menus proposés par la tablette sont décrits ci dessous :
 - Ecran_Lancement : écran affichant le nom de l'entreprise auteur de l'application ainsi que le nom de l'application.
 - Ecran_IdentificationTesteur : écran d'identification de l'opérateur.
 - Ecran_Principal : écran de menu principal laissant plusieurs choix d'actions à l'opérateur.
 - Ecran_StatistiqueTest : écran affichant les statistiques des derniers tests réalisés sur les cartes IPX800V4.
 - Ecran_HistoriqueCarte : écran affichant l'historique des tests ayant été effectués sur la carte IPX800V4 recherchée antérieurement.
 - Ecrans_Test : écran principal regroupant les différentes étapes du lancement d'un nouveau test dont :
 - Ecran_AttenteTest : écran affichant les tests en attente de démarrage.
 - Ecran_EditionTest : écran permettant de sélectionner les différents tests à effectuer sur la carte IPX800-v4 identifiée antérieurement.
 - Dialogue_ErreurEditionTest : écran de dialogue permettant d'informer l'opérateur qu'aucun test n'a été sélectionné dans l'écran Ecran_EditionTest.
 - Ecran_ProgressionTest : écran d'affichage de progression des tests en cours sur la carte IPX800V4 identifiée antérieurement.
 - Dialogue_ConfirmerInterruption : écran de dialogue demandant à l'opérateur de confirmer son intention de quitter un test en cours de réalisation.
 - Ecran_ResultatTest : écran permettant d'afficher les résultats du test venant d'être effectué sur une carte IPX800V4.
 - Dialogue_ConfirmerQuitter : écran de dialogue demandant à l'opérateur de confirmer son intention de quitter l'écran de résultats des tests.
 - Dialogue_ErreurInvalide : écran de dialogue informant l'opérateur de qu'il

souhaite quitter alors qu'il n'a pas validé ou invalidé les tests.

- Dialogue_TesteurDeconnexion : écran de dialogue demandant à l'utilisateur une confirmation de son intention de se déconnecter.
- Dialogue_RechercheCarte : dialogue de recherche de l'identifiant d'une carte.
- Dialogues_IdentificationCarte : dialogue d'identification de la carte IPX800V4 à tester.
 - Dialogue_IdNewCarte : dialogue d'identification d'une nouvelle carte.
 - Dialogue_IdCarte : dialogue d'identification d'une carte déjà existante.
- Dialogue_ErreurIdCarte : dialogue d'erreur lors de la saisie du numéro d'une carte déjà testée.
- Dialogue_CréationRecherche : dialogue de chargement avant l'affichage du menu de test.
- Dialogue_ErreurConnexion : écran de dialogue permettant d'indiquer que la communication a été interrompue.
- Dialogue_ReconnecterOuQuitter : écran de dialogue permettant de demander à l'utilisateur s'il souhaite quitter ou tenter une reconnexion suite à un problème de connexion.

2.2.2.2.3 En provenance de l'IPX800V4

2.2.2.2.3.1 Vers Thingy

Voici l'évènement logique en provenance de l'IPX800V4 vers Thingy :

- reponse(test) : retourne les résultats du test demandé antérieurement par Thingy. Les tests disponibles sont les mêmes que ceux décrits au paragraphe 2.2.2.2.4.1 Depuis Thingy. La réponse à ces tests peut être :
 - testSUCCEED
 - testFAIL
- retourDataEnocean : retourne les données reçues par l'IPX800V4 (trame Enocean envoyée par la X8R).
- reponsePing() : retourne la réponse de l'IPX800V4 à une requête de type « ping » envoyée antérieurement par Thingy

2.2.2.2.3.2 Vers E_X8R

Voici l'évènement logique en provenance de l'IPX800V4 vers E_X8R :

- envoyerEtatSortiesRelais(n) : retourne l'état des sorties relais de l'IPX800V4. Cet état peut-être 0 ou 1 et n, correspondant au numéro de la sortie relais envoyée, varie de 1 à 8.

2.2.2.2.4 A destination de l'IPX800V4

2.2.2.2.4.1 Depuis Thingy

Voici les événements logiques en provenance de Thingy vers l'IPX800V4 :

- demandeDataEnocean() : demande à l'IPX800V4 de retourner les données qu'elle a reçu de E_X8R sous la forme d'une trame Enocean.
- ordre(test) : demande à l'IPX800V4 d'effectuer un test particulier.
- envoyerPing() : envoie une requête de type « ping » à l'IPX800V4

Les tests disponibles (test) en provenance de Thingy à destination de l'IPX800V4 (ou en provenance de l'IPX800V4 à destination de Thingy) sont les suivants :

- test_entreesDigitales(n) : test des entrées logiques de l'IPX800V4 (n correspond au nombre d'entrées logiques et varie de 1 à 4).
- test_entreesAnalogiques(n) : test des entrées aalogiques de l'IPX800V4 (n correspond au nombre d'entrées logiques et varie de 1 à 4).
- test_sortiesRelais(n) : test des sorties relais de l'IPX800V4 (n correspond au nombre d'entrées logiques et varie de 1 à 8).
- test_busTerrain2Fils : test du bus de communication 2 fils.
- test_connectiviteEthernet : test de la connectivité Ethernet.
- test_memoireFlash : test de la mémoire Flash.
- test_memoireNVRAM : test de la mémoire NVRAM.

Voici la répartition de ces tests :

Test	Vérification via...
test_entreesDigitales	E_X8R
test_entreesAnalogiques	E_X8R
test_sortiesRelais	E_X8R
test_busTerrain2Fils	E_X8R
test_connectiviteEthernet	Thingy_Hard
test_memoireFlash	Thingy_Hard
test_memoireNVRAM	Thingy_Hard

2.2.2.2.4.2 Depuis E_X8R

Voici l'événement logique en provenance de E_X8R vers l'IPX800V4 :

- envoyerEtatSortiesRelais(n) : retourne l'état des sorties relais de la X8R. Cet état peut être 0 ou 1 et n, correspondant au numéro de la sortie relais envoyée, varie de 1 à 4.
- envoyerTrameEnocean(data) : envoi de données (data) par une trame Enocean à l'IPX800V4.

2.2.2.2.5 En provenance du serveur de tests

2.2.2.2.5.1 Vers Thingy

Voici les événements logiques en provenance du serveur de tests vers Thingy :

- validerIdentifiantsOperateur : signale que les identifiants de l'opérateur sont valides ou non
- validerVerificationIdentifiantCarte : signale que les identifiants de la carte sont valides ou non
- retourDonneesCarte : retourne les informations relatives à une carte (identifiant, tests effectués, opérateur)
- retourStatistiques : retourne les informations de statistiques de l'ensemble des derniers tests

2.2.2.2.6 A destination du serveur de tests

2.2.2.2.6.1 Depuis Thingy

Voici les événements logiques en provenance de Thingy vers le serveur de tests :

- demanderVerificationIdentifiantsOperateur : demande la recherche de la valeur d'un identifiant d'opérateur dans la base de données
- demanderVerificationIdentifiantsCarte : demande la recherche de la valeur d'un identifiant de carte dans la base de données
- demanderDonneesCarte : demande les informations relatives à une carte (identifiant, tests effectués, opérateur)
- demanderStatistiques : demande les informations de statistiques de l'ensemble des derniers tests
- demanderEnregistrementNouvelIdentifiant : demande l'enregistrement d'un nouvel identifiant d'une carte
- demanderEnregistrementTest : demande l'enregistrement d'un test venant d'être effectué sur une carte
- demanderEnregistrementValidation : demande l'enregistrement de la validation ou de l'invalidation d'une suite de tests venant d'être effectuée sur une carte

Seconde interface logique envisagée : utilisation de TE

L'Illustration 5 (page 26) présente le contexte de Thingy en faisant figurer les entrées/sorties dites de haut niveau (ou logiques). Elles sont regroupées en grande famille. Pour représenter ce contexte logique, un diagramme de communication UML a été utilisé. Nous retrouvons les périphériques Thingy déjà présentés en Illustration 3 (page 17).

Contexte logique du SaE

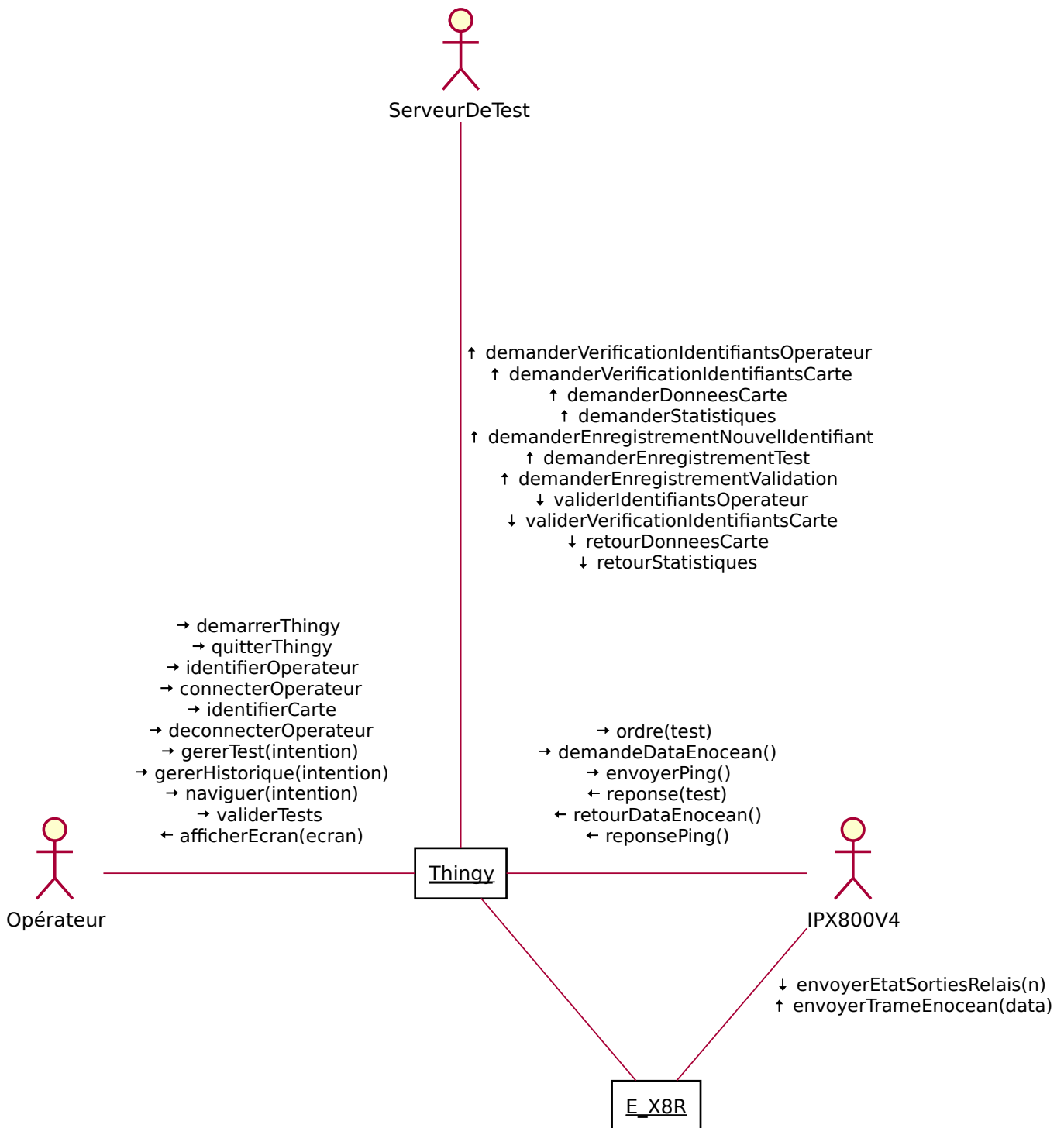


Illustration 5: Contexte logique du SaE avec TE

Dans ce diagramme de communication, seules les entrées/sorties logiques entre les acteurs et les périphériques Thingy sont présentées.

2.2.2.2b Les interfaces avec les acteurs

Les interactions entre les acteurs et Thingy varient seulement pour les communications entre E_X8R et l'opérateur ou l'IPX800V4. Pour les autres interactions, elles sont en tout point similaires à celles présentées au paragraphe précédent Première interface logique envisagée .

Nous allons maintenant détailler ces modifications.

2.2.2.2b.1 A destination de l'IPX800V4

2.2.2.2b.1.1 Depuis Thingy

Voici les événements logiques en provenance de Thingy vers l'IPX800V4 :

- demandeDataEnOcean() : demande à l'IPX800V4 de retourner les données qu'elle a reçu de E_X8R sous la forme d'une trame EnOcean.
- ordre(test) : demande à l'IPX800V4 d'effectuer un test particulier.
- envoyerPing() : envoie une requête de type « ping » à l'IPX800V4

Les tests disponibles (test) en provenance de Thingy à destination de l'IPX800V4 (ou en provenance de l'IPX800V4 à destination de Thingy) sont les suivants :

- test_entreesDigitales(n) : test des entrées logiques de l'IPX800V4 (n correspond au nombre d'entrées logiques et varie de 1 à 4).
- test_entreesAnalogiques(n) : test des entrées analogiques de l'IPX800V4 (n correspond au nombre d'entrées logiques et varie de 1 à 4).
- test_sortiesRelais(n) : test des sorties relais de l'IPX800V4 (n correspond au nombre d'entrées logiques et varie de 1 à 8).
- test_busTerrain2Fils : test du bus de communication 2 fils.
- test_connectiviteEthernet : test de la connectivité Ethernet.
- test_memoireFlash : test de la mémoire Flash.
- test_memoireNVRAM : test de la mémoire NVRAM.

Certains de ces tests ne sont pas réalisables en utilisant uniquement la carte X8R comme extension. C'est pourquoi, une carte électronique, Thingy_Electronisk, a été ajoutée au sein de Thingy_Hard (voir chapitre 2.2.1 Architecture matérielle et logicielle).

Voici la répartition de ces tests :

Test	Vérification via...
test_entreesDigitales	Thingy_Elektronisk
test_entreesAnalogiques	Thingy_Elektronisk
test_sortiesRelais	E_X8R
test_busTerrain2Fils	E_X8R
test_connectiviteEthernet	Thingy_Rasp
test_memoireFlash	Thingy_Rasp
test_memoireNVRAM	Thingy_Rasp

2.2.2.2b.2.2 Depuis E_X8R

Voici l'événement logique en provenance de E_X8R vers l'IPX800V4 :

- envoyerTrameEnocean(data) : envoi de données (data) par une trame EnOcean à l'IPX800V4.

2.2.2.3 Les interfaces physiques

Ce paragraphe précise les caractéristiques de chaque interface entre le logiciel et les composants matériels du système. Il s'agit des entrées/sorties bas-niveaux (dites aussi physiques). Ce sont celles que devra réellement traiter le SaE en les interprétant ou les générant en événement logique. Cela comprend aussi les caractéristiques de configuration (nombre de ports, jeux d'instruction, etc.), les contraintes électriques...

Deux interfaces physiques seront présentées dans ce paragraphe. En effet, l'une présentera la solution initiale proposée par le client tandis que l'autre intégrera au système une carte électronique conçue dans le but de préciser la réalisation des tests sur la carte IPX800V4.

En fonction du temps nécessaire à la conception et aux tests fonctionnels de la carte électronique, l'une ou l'autre des ces deux interfaces physiques sera choisie pour la concrétisation du projet.

Première interface physique envisagée : pas d'utilisation de TE

La suite de ce chapitre décrit chacune de ses interfaces. L'illustration 6 (page 29) représente ce contexte physique avec un diagramme de communication UML.

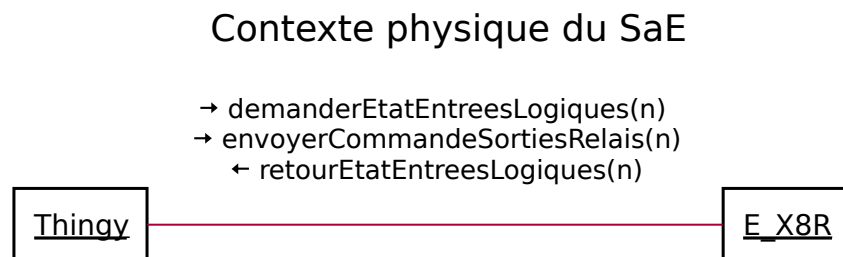


Illustration 6: Contexte physique du SaE sans TE

2.2.2.3.1.a E_X8R

La carte E_X8R est relié à Thingy_Hard par un câble dongle USB RS485.

La carte E_X8R va recevoir les événements suivants :

- demanderEtatEntreesLogiques(n) : envoi d'un signal permettant d'accéder à l'état des entrées logiques de la X8R. Cet état peut être 0 ou 1 et n, correspondant au nombre d'entrées logiques de la X8R, varie de 1 à 8. Le

protocole de communication utilisé pour l'envoi de ce signal est décrit au chapitre 2.2.2.5 Les interfaces de communication.

- envoyerCommandeSortiesRelais(n) : envoi d'un signal permettant d'accéder de modifier l'état des sorties relais de la X8R. Cet état peut être 0 ou 1 et n, correspondant au nombre de sortie relais de la X8R, varie de 1 à 8. Le protocole de communication utilisé pour l'envoi de ce signal est décrit au chapitre 2.2.2.5 Les interfaces de communication.

La carte E_X8R va transmettre les informations suivantes :

- retourEtatEntreesLogiques(n) : envoi d'un signal de réponse correspondant à l'état d'une entrée logique de la X8R. Cet état peut être 0 ou 1 et n, correspondant au nombre d'entrées logiques de la X8R, varie de 1 à 8. Le protocole de communication utilisé pour la réception de ce signal est décrit au chapitre 2.2.2.5 Les interfaces de communication.

Tous les protocoles de communication internes au produit Thingy (protocoles cités au chapitre 2.2.2.1 Les Interfaces logiques, notamment sur l'Illustration 4) seront décrits et explicités au chapitre 2.2.2.5 Les interfaces de communication.

Seconde interface physique envisagée : utilisation de TE

Contexte physique du SaE

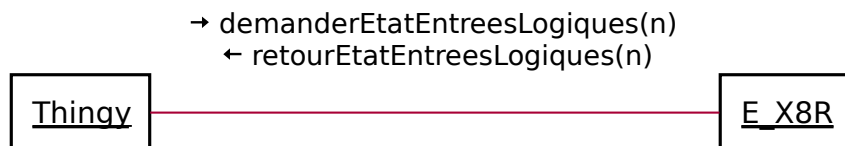


Illustration 7: Contexte physique du SaE avec TE

2.2.2.3.1.b E_X8R

La carte E_X8R est relié à Thingy_Hard par un câble dongle USB RS485.

La carte E_X8R va recevoir les évènements suivants :

- demanderEtatEntreesLogiques(n) : envoi d'un signal permettant d'accéder à l'état des entrées logiques de la X8R. Cet état peut être 0 ou 1 et n, correspondant au nombre d'entrées logiques de la X8R, varie de 1 à 8. Le protocole de communication utilisé pour l'envoi de ce signal est décrit au chapitre 2.2.2.5 Les interfaces de communication.

La carte E_X8R va transmettre les informations suivantes :

- retourEtatEntreesLogiques(n) : envoi d'un signal de réponse correspondant à l'état d'une entrée logique de la X8R. Cet état peut être 0 ou 1 et n, correspondant au nombre d'entrées logiques de la X8R, varie de 1 à 8. Le protocole de communication utilisé pour la réception de ce signal est décrit au chapitre 2.2.2.5 Les interfaces de communication.

Tous les protocoles de communication internes au produit Thingy (protocoles cités au chapitre 2.2.2.1 Les Interfaces logiques, notamment sur l'Illustration 4) seront décrits et explicités au chapitre 2.2.2.5 Les interfaces de communication.

2.2.2.4 Les interfaces avec les logiciels

Il n'est pas prévu que Thingy communique avec d'autres logiciels autres que ceux prévus pour les activités de développement de Thingy_Soft et Thingy_Hard.

2.2.2.5 Les interfaces de communication

Dans ce paragraphe sont définies les interfaces de communication (et les protocoles de communication associés).

Les différentes interfaces de communication sont les suivantes :

- Entre TH et IPX800V4
- Entre TH et X8R
- Entre IPX800V4 et X8R
- Entre TH, TA & PC de développement

2.2.2.5.1 Entre TH et IPX800V4

La communication entre ces 2 éléments se fait via une connexion ethernet entre eux. Cette communication se fera dans les 2 sens, et seront soumis à un protocole TCP/IP pour l'utilisation de trames TCP/IP ayant pour but, de pouvoir piloter l'IPX800V4 depuis TH via l'API [API-IPX800V4] fournie par GCE Electronics.

L'API fournit différentes fonctions, mais seules les suivantes seront employées par Thingy et envoyées sous la forme de requêtes URL :

- SetR : La commande SetR a pour action de mettre à 1 la sortie qui sera demandée.
 - Exemple : SetR = 04 met le relais 04 à l'état 1
- ClearR : La commande ClearR a pour action de mettre à 0 la sortie qui sera demandée.
 - Exemple : ClearR = 04 met le relais 04 à l'état 0
- Get : La commande Get permet d'obtenir l'état des entrées/sorties demandées
 - Exemple :
 - Get = R renvoie l'état de tous les relais

2.2.2.5.2 Entre TH et X8R

La communication entre TH et la carte X8R modifiée se fait par un dongle TTL-USB 115-200 Baud.

Les commandes utilisées pour piloter les sorties relais de la X8R sont de la forme : « **->RLYO=Sn** » en ASCII où « n » représente le caractère de fin 0x0A en hexadécimal, O correspond au numéro de la sortie relai à piloter et S correspond à la valeur (1 ou 0) à attribuer à la sortie relais considérée.

La X8R, de son côté, nous renvoie un signal de la forme suivante dès qu'une de ses entrées numériques ou sorties relais est modifiée :

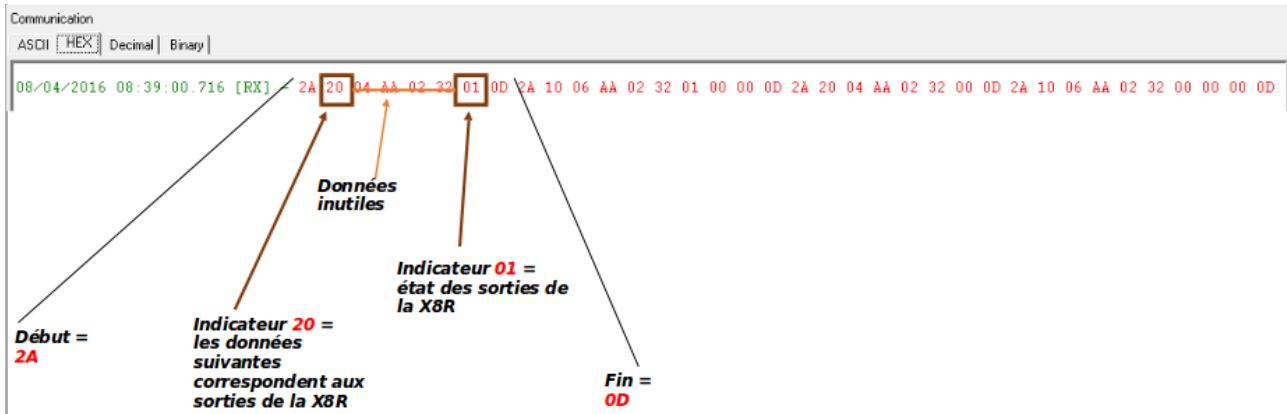


Illustration 8: Capture d'écran de l'envoi d'un signal de la X8R

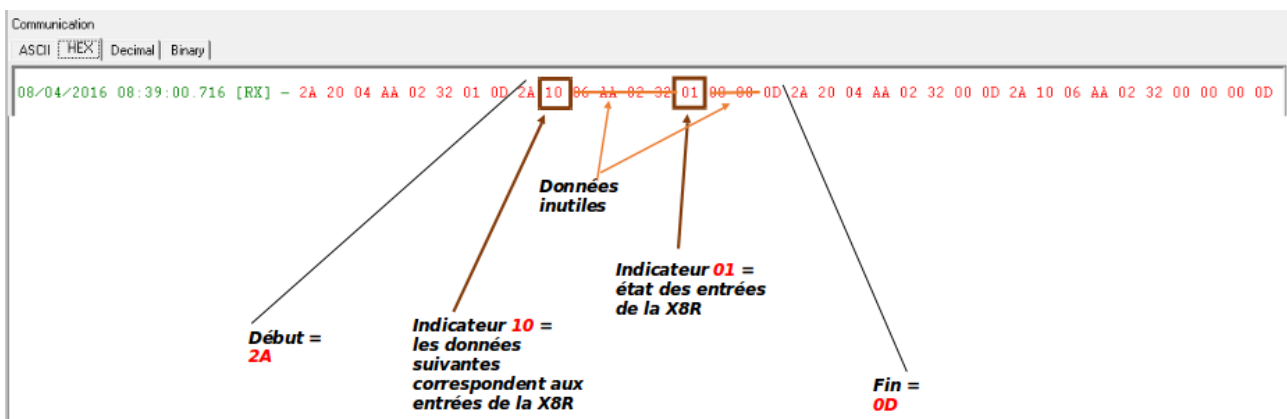


Illustration 9: Capture d'écran de l'envoi d'un signal de la X8R

Pour l'élaboration de notre produit, seule la lecture des entrées de la X8R sera utilisée.

Voici un tableau récapitulatif des valeurs renvoyées par la X8R en cas de changement d'une entrée numérique donnée.

Numéro de l'entrée numérique modifiée	Indicateur (en hexadécimal)	
	Mise à 1	Mise à 0
1	01	00
2	02	
3	04	
4	08	
5	10	
6	20	
7	40	
8	80	

2.2.2.5.3 Entre IPX800V4 et X8R

La communication entre ces deux composants se fait via un bus de communication (bus de terrains deux fils). Cette communication est totalement transparente pour le SaE.

(La X8R est alimentée par un câble RJ12 relié à l'IPX800V4).

2.2.2.5.4 Entre TH, TA & PC de développement

2.2.2.5.4.1 Entre TH & TA

La communication entre TH & TA se fait via WiFi. La communication s'effectuera via Sockets avec le protocole TCP/IP.

2.2.2.5.4.2 Entre TH & PC de développement

L'accès à TH via le PC de développement se fera par un protocole telnet/ssh.

2.2.3 Les contraintes de mémoire

Dans le cadre de la réalisation du produit Thingy, un prototype d'une base de donnée (nommé ServeurDeTest) sera élaboré.

A terme, ce prototype devra pouvoir aboutir à la création d'une base de données contenant l'historique de l'ensemble des tests réalisés sur des cartes IPX800V4 pendant une année complète, soit environ 10 000 tests.

2.2.4 Contraintes matérielles

La principale contrainte matérielle imposée par le projet de réalisation du produit Thingy est bien sûr l'utilisation d'une carte IPX800V4, carte électronique faisant l'objet principal du projet. Les contraintes d'utilisation d'une carte X8R sont décrites en dernière page du cahier des charges fourni par le client. [CDCProSe_GCEv2_SANS_EA2]

2.2.5 Exigences d'adaptation

Thingy, et plus précisément Thingy_App, devra être adaptable pour une utilisation sur une tablette Android 7" dont la version sera à minima 4.2.

Une possibilité d'évolution du produit et de ses fonctionnalités serait l'adaptabilité de Thingy_App à un écran de smartphone Android.

2.3 Fonctions principales développées

Ce chapitre présente les fonctionnalités principales développées dans l'incrément 1 en utilisant une démarche par cas d'usage et par Cas d'Utilisation (CU).

2.3.1 Rappel sur les cas d'usage

Les cas d'usage recensent les étapes essentielles du cycle de vie d'un produit depuis la fabrication du produit jusqu'à l'élimination ou recyclage de ce produit. A chaque étape du cycle de vie correspond un cas d'usage (si cette étape induit des fonctionnalités à définir pour le produit considéré). Pour chaque cas d'usage, plusieurs cas d'utilisations distincts peuvent être définis.

Parmi les cas d'usage, on retrouve généralement ceux de fabrication du produit (comprenant ou non les activités de test du produit fabriqué), de conditionnement (paramétrage éventuel, expédition et transport), de commercialisation (paramétrage éventuel, mode de démonstration, installation sur site...), d'utilisation (par le ou les utilisateurs), de maintenance (SAV ou diagnostique), de désinstallation et de recyclage (élimination ou valorisation).

2.3.2 Rappel sur les cas d'utilisation

Un Cas d'Utilisation (CU) représente un ensemble d'interactions entre un ou des acteurs et le système à spécifier. Ce cas d'utilisation étant souvent lié à un ou parfois plusieurs cas d'usage. Un CU est principalement décrit par un scénario d'utilisation (nommé scénario nominal), scénario d'une utilisation représentative du système. Ces CU sont ensuite détaillés jusqu'à un niveau de décomposition suffisant pour décrire les fonctions attendues du système.

2.3.2.1 Représentation graphique des CUs

Les CU peuvent être représentés sous forme graphique, voir l'Illustration 10 (page 37) pour une illustration. Les acteurs directs sont représentés sous forme de petits personnages. Dans les bulles sont représentés les cas d'utilisation. Un trait entre un acteur et un CU indique que l'acteur participe à ce CU. Les liens hachurés entre CU, préfixés par le mot <<use>> (ou <<include>>), indique que ce CU fait appel à l'autre CU – on parle alors de sous-cas d'utilisation. Les liens hachurés entre CU, préfixés par le mot <<extend>>, indique qu'il s'agit d'une extension d'un CU : un CU qui ne se déclenche que sous certaines conditions.

Principaux concepts d'un cas d'utilisation

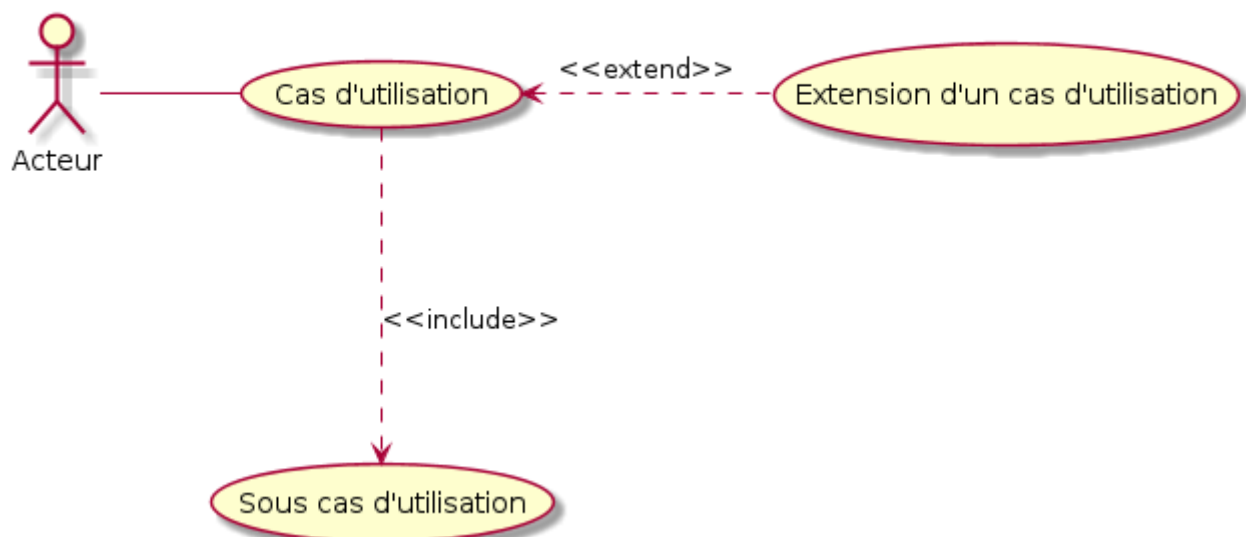


Illustration 10: Principaux concepts d'un cas d'utilisation

2.3.2.2 Représentation textuelle des CUs

La description textuelle des cas d'utilisation est souvent présentée sous forme d'un tableau, constitué des champs suivants :

Titre	Rappelle en quelques mots l'objectif principal du cas d'utilisation.
Résumé	Décrit brièvement le comportement du cas d'utilisation.
Portée	Définit la portée de conception du CU (étendue spatiale).
Niveau	Niveau de granularité du cas d'utilisation (Stratégique, utilisateur ou sous-fonction).
Acteur principal	Acteur qui déclenche le CU.
Acteur secondaire	Acteurs qui participent au CU.
Préconditions	Ensemble des conditions qui doivent être vérifiées avant le déroulement du CU. Les préconditions, sans mention contraire explicite, des CU parents au CU doivent toujours être vérifiées.
Garanties minimales	Définit ce qui est garanti par le système à l'étude même en cas d'échec du cas d'utilisation.
Garanties en cas de succès	Définit les garanties en cas de succès (par le scénario nominal ou par ses variantes).
Scénario nominal	C'est un scénario représentatif de l'utilisation du système ou tout se passe bien. Il se termine par la réussite des objectifs. Il est constitué d'une condition déclenchant le scénario, d'un ensemble d'étape, d'une condition de fin, et éventuellement d'extensions ou de variantes. Une étape peut être une interaction entre acteurs, une étape de validation, ou un changement interne.
Variantes	Lorsqu'il y a plusieurs façons de procéder à une même étape sans remise en cause du scénario nominal.
Extensions	Définissent les autres scénarios que le scénario nominal (par exemple ceux qui se terminent par un échec). Ils se déclenchent sur des conditions spécifiques détectées par le SaE.
Informations complémentaires	Informations diverses nécessaires à la compréhension du CU

2.3.3 Résumé des cas d'usage considérés pour Prøve

Les seuls cas d'usage considérés pour le SaE sont les suivants : démarrer Thingy, identifier l'opérateur, lancer un nouveau test, identifier la carte, configurer les tests, consulter les statistiques de tests, consulter l'historique des tests d'une carte, se déconnecter et quitter Thingy.

2.3.4 Résumé des cas d'utilisations stratégiques

L'illustration 11 (page 39) présente les cas d'utilisation (CU) principaux du SaE. Ils seront désignés par le terme de "CU stratégiques" car ils synthétisent les grandes utilisations de Prøve dans les cas d'usage traités dans ce document.

Cas d'utilisation stratégique de Thingy : Evaluer la fiabilité des cartes IPX800V4"

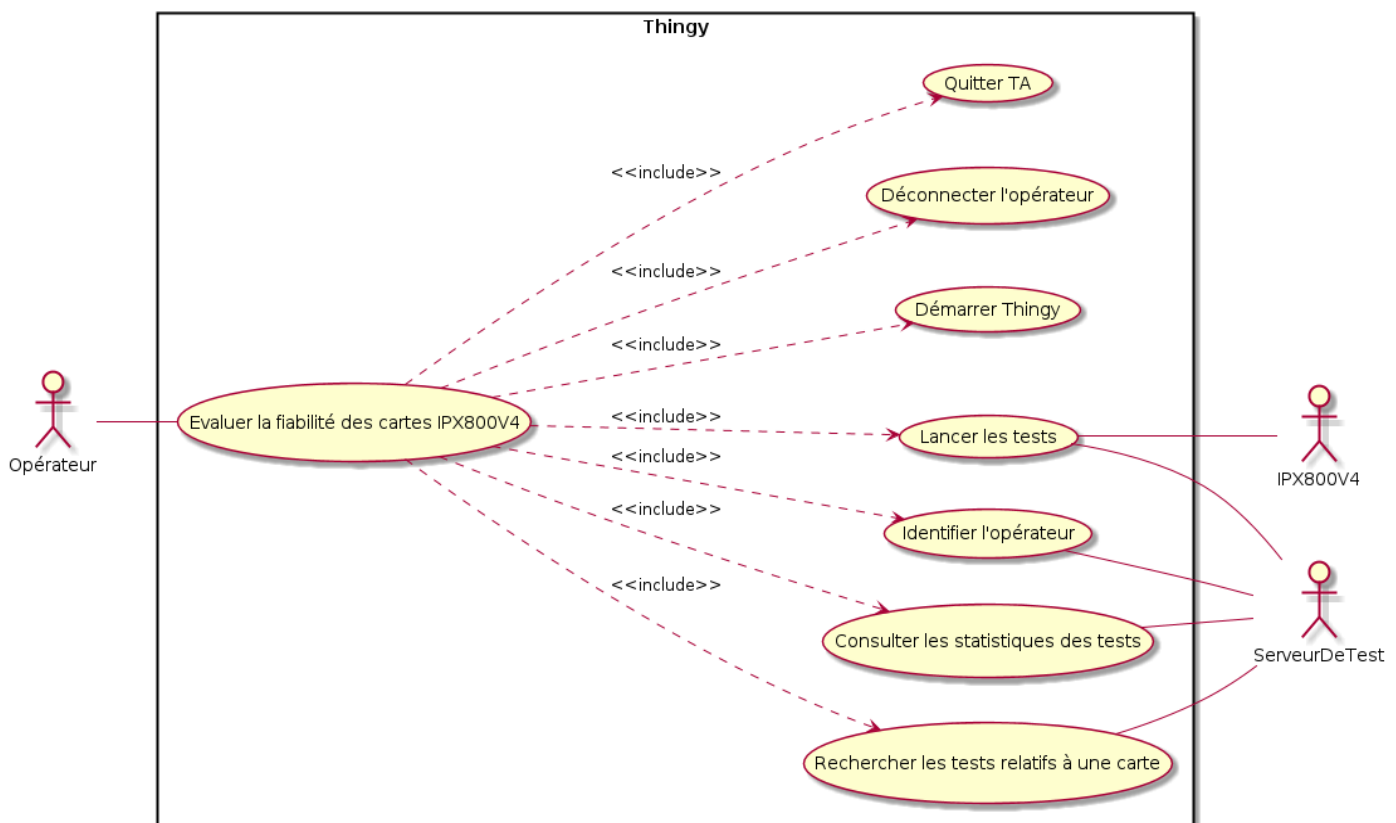


Illustration 11: CU stratégique Thingy

Les chapitres suivants vont détailler ces cas d'utilisation.

Titre	Évaluer la fiabilité des cartes IPX800V4
Résumé	L'opérateur évalue la fiabilité des cartes IPX800V4 en testant des cartes et en consultant leur historique
Niveau	Stratégique
Acteur principal	Opérateur
Acteur secondaire	IPX800V4, Serveur de test
Préconditions	Thingy est en état de fonctionner Thingy est connectée
Garanties minimales	
Garanties en cas de succès	L'opérateur a pu s'identifier. L'opérateur a pu lancer le test d'une carte.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur <u>démarre le SaE</u> 2. TA affiche l'écran d'identification de l'opérateur 3. L'opérateur <u>s'identifie</u> 4. Thingy récupère l'historique des derniers tests. 5. TA affiche l'écran principal 6. L'opérateur <u>lance un nouveau test</u>. 7. L'opérateur <u>quitte TA</u>.
Variantes	<p>6. [ConsulterStatistiques]</p> <p>6.1.a L'opérateur <u>consulte les statistiques des cartes</u>.</p> <p>6.1.b Va en 4)</p> <p>[RechercherTest]</p> <p>6.2.a L'opérateur <u>recherche les tests relatifs à une carte donnée dans la base de données</u>.</p> <p>6.2.b Va en 4)</p> <p>[Déconnecter]</p> <p>6.3.a L'opérateur <u>se déconnecte</u></p> <p>6.3.b Va en 2)</p> <p>[AffichageEcranPrincipal]</p> <p>6.4.a : Va en 4)</p> <p>7. [AffichageEcranIdentification]</p> <p>7.1.a : Va en 2)</p> <p>[AffichageEcranPrincipal]</p> <p>7.2.a : Va en 4)</p>

	1..2 : Va en 7) 3..5 : Va en 6)
Extensions	2..7 [Er_Com] : Perte de communication TC-TA [Er_Interrupt] : L'application s'interrompt de manière inattendue] [Er_Alim] : Thingy n'est plus alimenté]
Informations complémentaires	

2.3.5 CU_01 Démarrer Thingy

2.3.5.1 Description textuelle

Titre	Démarrer Thingy
Résumé	L'opérateur démarre le système à l'étude.
Acteur principal	Opérateur
Acteur secondaire	
Portée	Thingy
Préconditions	
Garanties minimales	
Garanties en cas de succès	<ul style="list-style-type: none"> • Accès au menu d'identification de Thingy.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur démarre TC. 2. TC informe qu'il est en état de fonctionner. 3. L'opérateur démarre TA. 4. TA affiche l'écran splash pendant le temps de lancement.
Variantes	
Extensions	1-4 : [Er_Alim] 1-4.a.1. Abandon du CU. 1-4 : [Er_Interrupt] 1-4.b.1. Abandon du CU.

2.3.6 CU_02 Identifier l'opérateur

2.3.6.1 Description textuelle

Titre	Identifier l'opérateur
Résumé	L'opérateur s'identifie auprès de l'application.
Acteur principal	Opérateur
Acteur secondaire	Serveur de test
Portée	Thingy
Préconditions	Thingy est en état de fonctionner. Le serveur de test est connecté.
Garanties minimales	Alerte l'opérateur en cas d'échec de l'identification.
Garanties en cas de succès	<ul style="list-style-type: none"> • Identification de l'opérateur. • Accès au menu principal de Thingy.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur entre ses identifiants. 2. L'opérateur demande à se connecter. 3. Thingy récupère les identifiants de l'opérateur. 4. Thingy vérifie le format des identifiants. 5. Thingy demande une vérification auprès du serveur. 6. Le serveur de test confirme les identifiants. 7. Thingy demande le nom complet de l'opérateur identifié auprès du serveur. 8. Le serveur de test donne les informations relatives à l'opérateur identifié à Thingy. 9. Thingy accorde l'accès au menu principal et transmet les données relatives à l'opérateur.
Variantes	<p>4. [ErreurFormatOpérateur]</p> <p>4.a.1. Thingy ne reconnaît pas le format des identifiants.</p> <p>4.a.2. Thingy affiche un message d'information.</p> <p>4.a.3. Retour en 1.</p> <p>6 : [Er_IdTesteur]</p> <p>6.a.1. Thingy ne trouve pas de correspondance.</p> <p>6.a.2. Thingy affiche un message explicatif.</p> <p>6.a.3. Retour en 1).</p>
Extensions	<p>3-8 : [Er_Com]</p> <p>3-6.a. Abandon du CU.</p>

2.3.7 CU_03 Lancer les tests

2.3.7.1 Description graphique

Détail du cas d'utilisation "Lancer les tests"

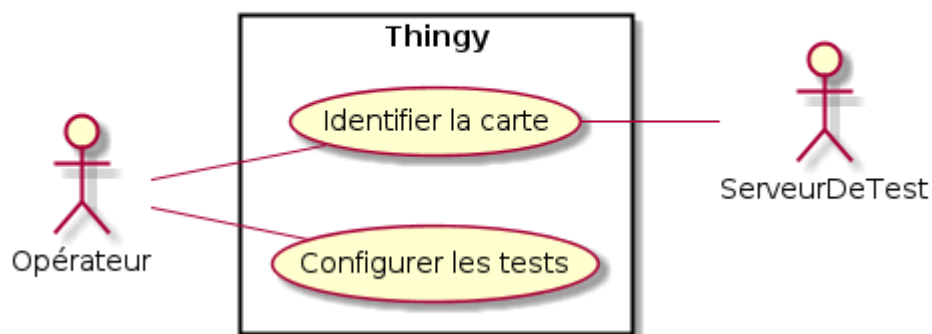


Illustration 12: CU LancerTests

2.3.7.2 Description textuelle

Titre	Lancer un nouveau test
Résumé	L'utilisateur lance un nouvel ensemble de tests sur une carte
Niveau	Utilisateur
Acteur principal	Opérateur
Acteur secondaire	IPX800V4, Serveur de test
Préconditions	L'opérateur s'est identifié.
Garanties minimales	Garantie de l'intégrité physique et logicielle du Thingy.
Garanties en cas de succès	<ul style="list-style-type: none">L'opérateur a pu tester l'IPX800V4.

Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur demande à lancer un nouveau test. 2. Thingy affiche un dialogue pour l'identification de la carte. 3. L'opérateur <u>identifie la carte</u>. 4. Thingy affiche l'écran de tests. 5. L'opérateur démarre le test. 6. Thingy affiche les résultats des différents tests tout en indiquant leur progression <ol style="list-style-type: none"> 6.1. de connectivité Ethernet. 6.2. du bus RJ12. 6.3. du bus de terrain 2 fils. 6.4. de la mémoire Flash. 6.5. de la mémoire NVRAM. 6.6. des sorties relais. 6.7. des entrées analogiques. 6.8. des entrées numériques. 7. Thingy enregistre les résultats des tests 8. Thingy affiche l'écran final de résultat des tests. 9. L'opérateur valide les résultats. 10. L'opérateur demande de quitter l'écran de résultats. 11. TC enregistre la validation des résultats dans le serveur de test. 12. Thingy demande l'historique des anciens tests auprès du serveur de test. 13. Le serveur de test retourne les historiques des anciens tests.
Variantes	<p>5. [ConfigurerTest]</p> <p>5.1.a L'opérateur <u>configure les tests</u>.</p> <p>5.1.b Va en 4)</p> <p>[QuitterEcranAttenteTest]</p> <p>5.2.a L'opérateur souhaite quitter l'écran d'attente de tests</p> <p>5.2.b Il annule sa demande.</p> <p>5.2.c Retour en 4).</p> <p>6. [ArretTestEnCours]</p> <p>6.1.a L'opérateur demande à quitter la réalisation des tests</p> <p>6.1.b Thingy demande une validation de l'arrêt des tests en</p>

	<p>cours</p> <p>6.1.c L'opérateur valide son choix</p> <p>6.1.d Va en 1)</p> <p>[AffichagePersonnalisé]</p> <p>6.2.a L'opérateur n'a pas choisit de lancer un tests intégral</p> <p>6.2.b Thingy n'affiche les résultats que des tests choisis</p> <p>6.2.c Va en 7)</p> <p>9. [ConsulterDétailTest]</p> <p>9.1.a L'opérateur consulte le détail des tests</p> <p>9.1.b Va en 7)</p> <p>[Invalidation]</p> <p>9.2.a L'opérateur invalide les résultats</p> <p>9.2.b L'opérateur demande de quitter l'écran de résultats.</p> <p>9.2.c TC enregistre l'invalidation des résultats dans le serveur de test.</p> <p>9.2.d Va en 10)</p> <p>10. [Replay]</p> <p>10.1.a L'opérateur demande un replay sur les tests.</p> <p>10.1.b TC enregistre la validation des résultats dans le serveur de test</p> <p>10.1.c Thingy demande l'historique des anciens tests auprès du serveur de test.</p> <p>10.1.d Le serveur de test retourne les historiques des anciens tests.</p> <p>10.1.e Va en 4)</p>
Extensions	<p>5.</p> <p>5.2.a L'opérateur souhaite quitter l'écran d'attente de tests</p> <p>5.2.b L'opérateur valide son choix.</p> <p>5.2.c Abandon du CU.</p>

2.3.8 CU_04 Identifier la carte

2.3.8.1 Description textuelle

Titre	Identifier la carte
Résumé	L'opérateur permet l'identification de la carte à tester.
Acteur principal	Opérateur
Acteur secondaire	Serveur de test
Portée	Thingy
Préconditions	Thingy est en état de fonctionner. L'opérateur est identifié. Le serveur de test est connecté.
Garanties minimales	Alerte l'opérateur en cas d'échec d'identification de la carte.
Garanties en cas de succès	<ul style="list-style-type: none"> • Identifier la carte à tester. • Accès à l'interface de test de Thingy.
Scénario nominal	<ol style="list-style-type: none"> 1. L'opérateur souhaite l'identification de la carte. 2. L'opérateur entre les identifiants de la carte. 3. L'opérateur valide son choix. 4. Thingy récupère les identifiants de la carte. 5. Thingy vérifie le format des identifiants. 6. Le serveur de test confirme les identifiants. 7. Thingy analyse le résultat obtenu. 8. Thingy donne l'accès au menu de test.
Variantes	<p>2. [NouvelleCarte]</p> <p>2.a.1. L'opérateur demande la création d'identifiants pour la carte.</p> <p>2.a.2 L'opérateur valide son choix.</p> <p>2.a.3. Thingy lance une requête de génération d'identifiants.</p> <p>2.a.4. Le serveur de test retourne les nouveaux identifiants.</p> <p>2.a.5. Retour en 7.</p> <p>5. [ErreurFormatCarte]</p> <p>5.a.1. Thingy ne reconnaît pas le format des identifiants.</p> <p>5.a.2. Thingy affiche un message d'information.</p> <p>5.a.3. L'opérateur informe qu'il a pris connaissance de cette information .</p> <p>5.a.4. Retour en 1.</p> <p>7 : [Er_IdCarte]</p> <p>7.a.1. Thingy ne trouve pas de correspondance.</p>

	7.a.2. Thingy affiche un message explicatif. 7.a.3. L'opérateur informe qu'il a pris connaissance de cette information . 7.a.4. Retour en 1).
Extensions	2-6 : [Er_Com] 2-6.a.1. Abandon du CU.

2.3.9 CU_05 Configurer les tests

2.3.9.1 Description textuelle

Ce CU permet de sélectionner ou désélectionner un ou plusieurs tests afin d'établir une liste de tests à exécuter (un test obligatoire ne peut pas être désélectionné).

Titre	Configurer les tests
Résumé	L'opérateur choisit entre plusieurs tests à effectuer.
Niveau	Utilisateur
Acteur principal	Opérateur
Acteur secondaire	
Préconditions	L'opérateur a sélectionné le bouton d'édition des tests.
Garanties minimales	
Garanties en cas de succès	L'opérateur a pu personnaliser les tests qu'il souhaite réaliser sur la carte à tester.
Scénario nominal	<ol style="list-style-type: none"> 1. TA affiche l'écran d'édition de tests. 2. L'opérateur choisit les différents tests qu'il souhaite réaliser sur l'IPX800V4 à tester. 3. L'opérateur valide ses choix.
Variantes	3. [Er_ChoixTest] 3.1.a Le SaE affiche un message d'erreur car un test obligatoire a été désélectionné. 3.1.b L'opérateur informe qu'il a pris connaissance de cette erreur 3.1.c Va en 1)
Extensions	1..3 : L'opérateur quitte l'édition de tests

2.3.10 CU_06 Consulter les rapports des anciens tests

2.3.10.1 Description textuelle

Titre	Consulter les rapports des anciens tests de la carte
Résumé	L'opérateur visualise les données relatives aux tests précédents de la carte recherchée
Acteur principal	Opérateur, serveur de test
Acteur secondaire	-
Portée	Thingy_Commande, Thingy_Affiche et Thingy_Hard
Préconditions	L'opérateur est identifié, la carte à tester est identifiée
Garanties minimales	Thingy s'allume
Garanties en cas de succès	<ul style="list-style-type: none">• Accès aux tests réalisés antérieurement sur la carte recherchée
Scénario nominal	<ol style="list-style-type: none">1. L'opérateur recherche un identifiant de carte.2. L'opérateur valide son choix.3. Thingy demande les données de la carte recherchée au serveur de test4. Le serveur de test retourne les données identifiées par Thingy5. Thingy affiche les résultats des anciens tests de la carte recherchée.6. L'opérateur quitte la recherche.
Variantes	<p>4. [Er_IdCarte]</p> <p>4.1.a. Le serveur de test ne trouve pas de carte associée à l'identifiant.</p> <p>4.1.b. Thingy affiche un message d'erreur</p> <p>4.1.c va en 6).</p> <p>5. [AbsenceEnregistrement]</p> <p>5.1.a Aucun test antérieur n'est enregistré.</p> <p>5.1.b Va en 6).</p>
Extensions	

2.3.11 CU_07 Consulter les statistiques

Ce CU est une projection et sera implémenté durant l'incrément 3.

2.3.11.1 Description textuelle

Titre	Consulter les statistiques
Résumé	L'opérateur visualise les résultats de l'ensemble des tests précédents
Acteur principal	Opérateur, serveur de test
Acteur secondaire	-
Portée	Thingy Soft et Thingy Hard
Préconditions	L'opérateur est identifié
Garanties minimales	Le SaE s'allume
Garanties en cas de succès	<ul style="list-style-type: none">• Accès aux statistiques de l'ensemble des tests effectués sur des cartes IPX800V4
Scénario nominal	<ol style="list-style-type: none">1. L'opérateur demande l'accès aux statistiques des tests.2. Thingy demande les données concernant les statistiques de tests auprès du serveur de test3. Le serveur de test retourne les données de statistiques des tests4. Thingy affiche l'écran de statistiques5. L'opérateur demande à retourner à l'écran principal
Variante:	
Extensions	

2.3.12 CU_09 Déconnecter Opérateur

2.3.12.1 Description textuelle

Titre	Déconnecter l'opérateur
Résumé	L'opérateur se déconnecte de Thingy.
Acteur principal	Opérateur
Acteur secondaire	
Portée	Thingy
Préconditions	Thingy est en état de fonctionner. L'opérateur est connecté.
Garanties minimales	
Garanties en cas de succès	<ul style="list-style-type: none">• L'opérateur s'est correctement déconnecté.
Scénario nominal	<ol style="list-style-type: none">1. L'opérateur souhaite se déconnecter.2. TA affiche un dialogue de déconnexion.3. L'opérateur valide la déconnexion.
Variantes	2. [AnnulerQuitter] 2.1.a L'opérateur souhaite annuler son intention de se déconnecter. 2.1.b Abandon du CU
Extensions	

2.3.13 CU_010 Quitter Thingy

2.3.13.1 Description textuelle

Titre	Quitter Thingy
Résumé	L'opérateur ferme le système à l'étude.
Acteur principal	Opérateur
Acteur secondaire	
Portée	Thingy
Préconditions	Thingy est en état de fonctionner.
Garanties minimales	
Garanties en cas de succès	<ul style="list-style-type: none">• Thingy s'est correctement arrêté.
Scénario nominal	<ol style="list-style-type: none">1. L'opérateur souhaite fermer Thingy.2. TA affiche un dialogue de fermeture de l'affichage.3. L'opérateur valide la fermeture.4. TA informe TC de son arrêt.5. TC s'arrête.6. TA s'arrête.
Variantes	3. [AnnulerQuitter] 3.1.a L'opérateur souhaite annuler son intention de quitter 3.1.b abandon du CU.
Extensions	

3. Exigences spécifiques

3.1 Interface Homme-Machine

3.1.1 Généralités

Le dispositif Thingy peut interagir avec l'IPXV800 V4 par bus et ports numériques ou analogiques. Thingy peut envoyer des informations à l'utilisateur par l'intermédiaire d'un écran situé sur la tablette dédiée à son usage.

Ce dossier de spécification prend en compte la possibilité de consulter des moyennes statistiques sur les tests déjà réalisés. Cette fonctionnalité ne sera implémentée que lors de l'incrément 3.

3.1.2 Vue générale

L'Illustration 13 (page 53) représente les navigations possibles entre les différents écrans et boîtes de dialogues proposés par l'IHM. Ces enchaînements sont représentés par des diagrammes d'état transition UML (Illustration 13, 14 et 15 page 53, 54 et 55).

Chaque écran ou boîte de dialogue est représenté par un état (rectangle arrondi). Les transitions entre les états (les flèches) représentent une navigation d'un état à l'autre en précisant la condition ou l'événement logique du contexte (cf. 2.2.2.1 Les Interfaces logiques) qui active la transition. Cela correspond à des actions faites par l'utilisateur sur les boutons virtuels de l'écran ou sur les périphériques de l'IPX800V4 pour générer l'événement correspondant. Les états enveloppant d'autres états sont des « états composites ». Toutes les transitions positionnées sur le bord de ces états composites s'appliquent à tous les états à l'intérieur.

Certaines transitions ne sont pas franchies sur des événements logiques, ce sont :

- Les événements de temporisation (le mot clef « after » est alors noté sur la transition) : la transition doit être sensibilisée pendant une certaine durée pour

être franchie.

- Des conditions qui deviennent vraies (la condition est alors exprimée entre crochets).

Voici ces diagrammes :

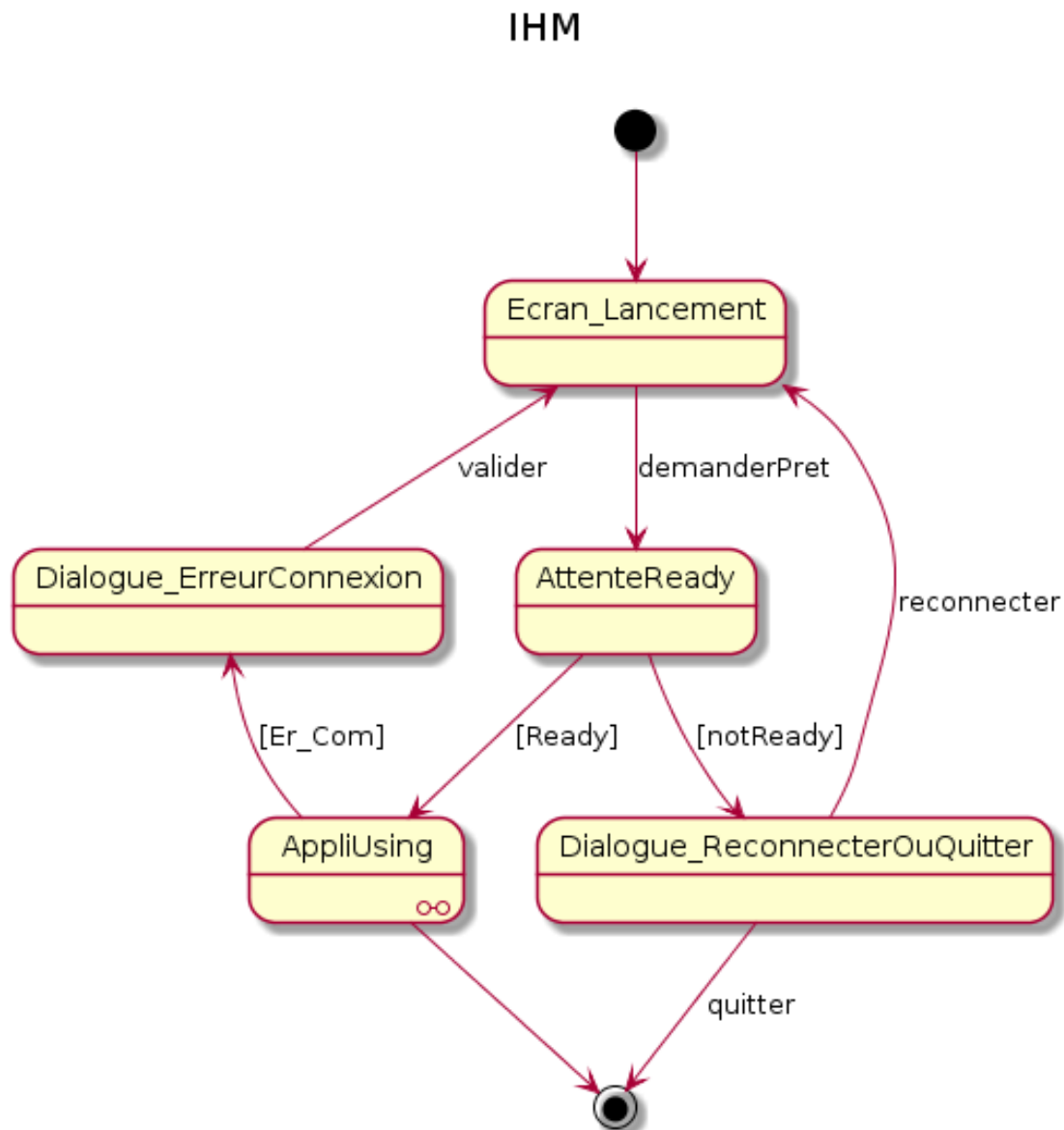


Illustration 13: IHM Intentions

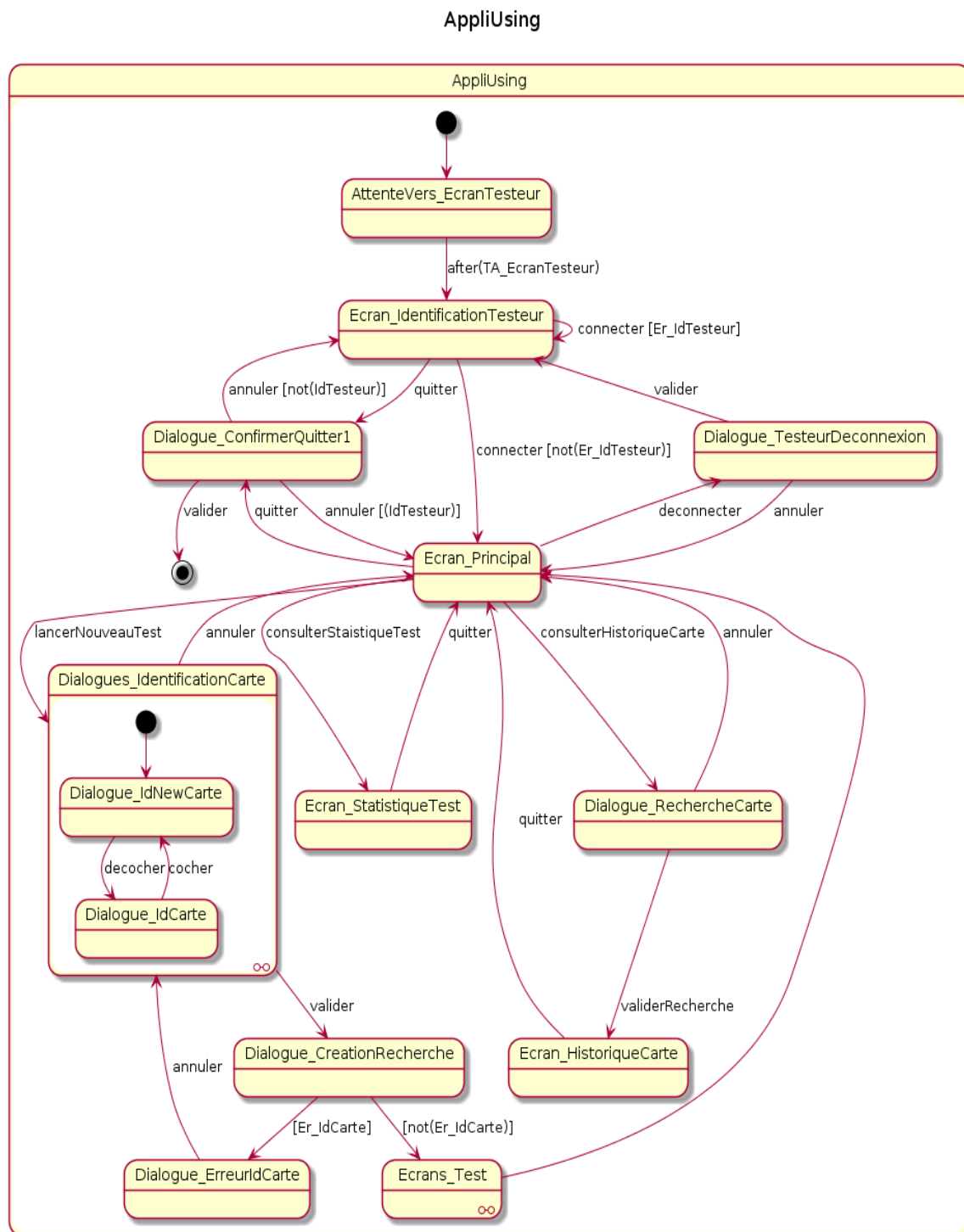


Illustration 14: IHM_AppliUsing

dialogue « Dialogue_ReconnecterOuQuitter » s'affiche. Depuis cette boîte de dialogue, l'utilisateur pourra soit tenter une reconnexion soit quitter. Si l'utilisateur tente une reconnexion, l'IHM retourne dans l'état « Ecran_Lancement ». Si l'utilisateur souhaite quitter, l'application se termine.

Si depuis l'état composite « AppliUsing » (et donc depuis tous ses sous états) une erreur de communication survient, l'utilisateur verra le « Dialogue_ErreurConnexion » s'afficher. Une fois que l'utilisateur aura fermé cette boîte de dialogue, l'IHM se retrouve dans l'état « Ecran_Lancement ».

Depuis le diagramme « AppliUsing » (Illustration 14) :

Au démarrage de cette machine à état, l'IHM se trouve dans l'état « AttenteVers_EcranTesteur ». L'utilisateur devra attendre un temps « TA_EcranTesteur » avant d'arriver sur l'écran « Ecran_IdentificationTesteur ».

A partir de l'écran « Ecran_IdentificationTesteur », l'utilisateur pourra :

- Saisir son identifiant et son mot de passe via un clavier virtuel affiché sur l'écran. Après validation, si l'identifiant et le mot de passe ne sont pas valides (Er_IdTesteur), l'utilisateur verra une phrase s'afficher sur cet écran pour lui indiquer que son identifiant ou son mot de passe comporte une erreur. Si à l'inverse l'identifiant et le mot de passe sont corrects et validés, l'utilisateur pourra passer à l'écran suivant : « Ecran_Principal ».
- Quitter l'application : une boîte de dialogue s'ouvrira « Dialogue_ConfirmerQuitter1 » qui demandera à l'utilisateur s'il est sûr de vouloir quitter l'application. S'il valide, l'application se termine. S'il annule, la boîte de dialogue se ferme.

A partir de l'écran « Ecran_Principal », l'utilisateur pourra :

- Lancer un nouveau test : pour cela, une boîte de dialogue « Dialogue_IdNewCarte » s'ouvre. Cette boîte de dialogue est un sous état de « Dialogues_IdentificationCarte ». A l'ouverture de cette boîte de dialogue, si l'utilisateur souhaite identifier une carte déjà existante, il n'aura qu'à décocher la case « créer une nouvelle carte » (cf. chapitre 3.1.4.2.1). Une nouvelle boîte de dialogue s'ouvre « Dialogue_IdCarte » qui est aussi un sous état de « Dialogues_IdentificationCarte » qui permettra à l'utilisateur de saisir un numéro de carte existant puis de valider. Si à l'inverse, l'utilisateur souhaite créer un nouvel identifiant de carte, il n'aura qu'à

cocher la case « créer une nouvelle carte » puis valider (cf. chapitre 3.1.4.3.2). Après validation, la boîte de dialogue s'ouvre « Dialogue_CreationRecherche » qui demande à l'utilisateur d'attendre la création du nouvel identifiant de carte ou d'attendre la recherche de la carte saisie. Si l'identifiant de la carte est incorrect (Er_IdCarte), la boîte de dialogue « Dialogue_ErreurIdCarte » s'ouvre et l'utilisateur devra fermer pour revenir à la boîte de dialogue « Dialogue_IdNewCarte ». Si à l'inverse, aucune erreur n'est survenue, l'écran « Ecran_AttenteTest » (sous état de « Ecrans_Test ») s'affiche.

- Consulter les statistiques des tests appliqués sur toutes les cartes IPX800 V4 : pour cela, l'écran « Ecran_StatistiqueTest » s'affichera. Il pourra revenir à l'écran « Ecran_Principal » en quittant cet écran.
- Consulter l'historique des tests appliqués sur une carte : la boîte de dialogue « Dialogue_RechercheCarte » s'affichera. L'utilisateur rentrera la carte pour laquelle il souhaite consulter l'historique. Après validation, l'écran « Ecran_HistoriqueCarte » s'affiche. L'utilisateur devra sélectionner le bouton quitter pour revenir à l'écran « Ecran_Principal ».
- Se déconnecter de son compte utilisateur : une boîte de dialogue s'ouvrira « Dialogue_TesteurDeconnexion » pour demander s'il est sûr de vouloir se déconnecter du compte identifié.
- Enfin, quitter l'application, une boîte de dialogue s'ouvrira « Dialogue_ConfirmerQuitter1 » qui demandera à l'utilisateur s'il est sûr de vouloir quitter l'application. S'il valide, l'application se termine. S'il annule, la boîte de dialogue se ferme.

Depuis le diagramme « Ecrans_Test » (Illustration 15) :

A partir de l'écran « Ecran_AttenteTest » l'utilisateur pourra :

- Editer le test à effectuer : l'écran « Ecran_EditionTest » s'affichera. Sur cet écran, l'utilisateur pourra cocher les tests qu'il souhaite effectuer ou décocher ceux qu'il ne veut pas effectuer. Si un test obligatoire a été décoché, une boîte de dialogue « Erreur_EditionTest » s'ouvrira pour annoncer à l'utilisateur qu'il doit effectués les tests obligatoires. Après avoir bien sélectionné les tests, l'utilisateur pourra valider sa configuration et il reviendra à l'écran « Ecran_AttenteTest ».

- Lancer le(s) test(s) (édité ou intégral) : Il suffira de sélectionner le bouton de lancement et l'écran « Ecran_ProgressionTest » s'affichera. où il devra attendre la fin du test pour pouvoir passer à l'écran suivant. Si l'utilisateur souhaite annuler la progression, une boîte de dialogue « Dialogue_ConfirmerInterruption » s'ouvrira pour demander à l'utilisateur s'il est sûr de vouloir interrompre les tests en cours. S'il confirme, l'écran « Ecran_AttenteTest » s'affichera. Sinon, la progression du test reprend. Une fois le(s) test(s) terminé(s), l'écran « Ecran_ResultatTest » s'affiche.

A partir de l'écran « Ecran_ResultatTest » l'utilisateur pourra :

- Valider ou invalider le(s) test(s) effectué(s). S'il n'a pas effectué cette action et qu'il souhaite relancer des tests ou revenir à l'écran « Ecran_Principal », la boîte de dialogue « Dialogue_ErreurInvalide » s'affichera pour lui indiquer qu'il doit valider ou invalider le(s) test(s) effectué(s).
- Quitter pour revenir à l'écran « Ecran_Principal » : une boîte de dialogue « Dialogue_ConfirmerQuitter2 » s'ouvrira pour demander à l'utilisateur s'il est sûr de vouloir revenir au menu principal.
- Rejouer des tests : l'écran « Ecran_AttenteTest » s'affichera.

3.1.3 Les actions utilisateur

Les différentes actions et intentions de l'utilisateur seront détaillées dans cette partie :

3.1.3.1 Depuis la vue générale (Illustration 13)

- **Depuis Ecran_Lancement**

Attendre le temps nécessaire pour le démarrage de l'application.

- **Depuis Dialogue_ConfirmerQuitter**

*valider : quitter l'application.

*annuler : revenir à l'écran :

« Ecran_Principal » si le testeur a été identifié (ID_Testeur) ou

« Ecran_IdentificationTesteur » si le testeur n'a pas été identifié.

- **Depuis Dialogue_ErreurConnexion**

*valider : fermer la boîte de dialogue et revenir à l'état « Ecran_Lancement ».

- **Depuis Dialogue_ReconnecterOuQuitter**

*reconnecter : fermer la boîte de dialogue et tenter une reconnexion du système et revenir à l'état « Ecran_Lancement ».

*quitter : fermer la boîte de dialogue terminer le système.

3.1.3.2 Depuis l'état « AppliUsing » (Illustration 14)

- **Depuis Ecran_IdentificationTesteur**

*entrerIdentifiantTesteur : rentrer l'identifiant par une saisie sur le clavier virtuel affiché sur l'écran.

*entrerCodeTesteur : rentrer le mot de passe associé à l'identifiant par une saisie sur le clavier virtuel affiché sur l'écran.

*validerTesteur : permet de :

Passer à l'écran « Ecran_Principal » si l'identifiant et le mot de passe du testeur ont bien été rentré ou

Rester sur cet écran et avoir un message d'erreur si l'identifiant ou le mot de passe est incorrect (Er_IdTesteur).

*quitter : quitter l'application.

- **Depuis Ecran_Principal**

*consulterHistoriqueCarte : emmène sur la boîte de dialogue « Dialogue_RechercheCarte » pour ensuite pouvoir consulter l'historique d'une carte.

*lancerNouveauTest : emmène sur la boîte de dialogue « Dialogue_IdNewCarte » afin de pouvoir saisir ou créer l'identifiant de la carte à tester.

*consulterStatistiqueTest : consulter les statistiques des tests appliqués à l'ensemble des cartes.

*deconnecter : revenir à l'écran « Ecran_IdentificationTesteur ».

*quitter : quitter l'application.

- **Depuis Ecran_HistoriqueCarte**

*quitter : retourner à l'écran « Ecran_Principal ».

- **Depuis Ecran_StatistiqueTest**

*quitter : retourner à l'écran « Ecran_Principal ».

- **Depuis Dialogue_TesteurDeconnexion**

*valider : confirmer la déconnexion de l'utilisateur et revenir à l'écran

« Ecran_IdentificationTesteur ».

*annuler : ne pas confirmer la déconnexion de l'utilisateur et revenir à l'écran

« Ecran_MenuPrincipal ».

- **Depuis Dialogue_RechercheCarte**

*saisirIdentifiantCarte : rentrer l'identifiant de la carte sur laquelle sera consulté l'historique des tests.

*valider: aller à l'écran : « Ecran_HistoriqueCarte ».

*annuler : revenir à l'écran précédent : « Ecran_Principal ».

- **Depuis Dialogue_CreationRecherche**

*Attendre la recherche ou la création d'un identifiant d'une carte.

- **Depuis Dialogue_ErreurIdCarte**

*annuler : fermer la boîte de dialogue « Dialogue_ErreurIdCarte » et revenir à la boîte de dialogue « Dialogue_IdNewCarte ».

3.1.3.3 Depuis « Dialogues IdentificationCarte »

- **Depuis Dialogue_IdNewCarte**

*decocher : décocher la case permet de pouvoir rentrer un identifiant de carte existant (envoie à la boîte de dialogue « Dialogue_IdCarte »).

*valider : valider la génération d'un nouvel identifiant de la carte à tester.

*annuler : revenir à l'écran « Ecran_Principal ».

- **Depuis Dialogue_IdCarte**

*cocher: cocher la case permet de pouvoir générer un nouvel identifiant de carte (envoie à la boîte de dialogue « Dialogue_IdNewCarte »).

*saisirIdentifiantCarte : rentrer l'identifiant de la carte sur laquelle seront effectués les tests.

*valider : valider la saisie de l'identifiant de la carte à tester.

*annuler : revenir à l'écran « Ecran_Principal ».

3.1.3.5 Depuis « Ecrans_Test » :

- **Depuis Ecran_AttenteTest**

*demarrerTest : démarrer le test (édité ou non) sur l'IPX800V4 identifié.

*editerTest : choisir quel(s) test(s) effectuer sur l'IPX800V4 identifié.

*quitter : revenir à l'écran « Ecran_Principal ».

- **Depuis Ecran_EditionTest**

*validerEdition : valider l'édition des tests.

*cocher : cocher un test pour qu'il soit exécuté.

*decocher : décocher un test pour qu'il ne soit pas exécuté.

*annuler : annuler l'édition des tests et revenir à la configuration précédente

- **Depuis Ecran_ProgressionTest**

*Attendre que la condition Test_Fini soit vraie.

*annuler : annuler le(s) test(s) en cours.

- **Depuis Ecran_ResultatTest**

*validerResultat : valider le résultat du test qui vient d'être exécuté.

*invaliderResultat : invalider le résultat du test qui vient d'être exécuté.

*rejouerTest : relancer des tests sur la carte : revenir à l'écran
« Ecran_AttenteTest ».

*quitter : revenir à l'écran « Ecran_Principal ».

- **Depuis Dialogue_ConfirmerQuitter**

*valider : quitter l'application.

*annuler : revenir à l'écran :

« Ecran_ResultatTest » si le test vient d'être terminé (Test_Fini) ou

« Ecran_AttenteTest » si le test n'a pas été lancé.

- **Depuis Dialogue_ConfirmerInterruption**

*valider : annule le test en cours.

*annuler : reprend le test en cours.

- **Depuis Dialogue_ErreurEditionTest**

*annuler : revenir à l'écran « Ecran_EditionTest ».

- **Depuis Dialogue_ErreurInvalide**

*annuler : revenir à l'écran « Ecran_ResultatTest ».

3.1.4 Les écrans et dialogues

Chacun des écrans et boîtes de dialogues vont être détaillés dans le chapitre suivant. Pour chaque écrans possédant des boutons virtuels, un tableau explicatif des boutons avec l'intention ou l'action associée sera ajouté.

3.1.4.1 Depuis la vue générale (Illustration 13)

3.1.4.1.1 Ecran_Lancement

Cet écran correspond au premier écran affiché au lancement de l'application. Cet écran affiche le nom du produit « Thingy » et les architectes du produit « Prøve ». L'utilisateur devra attendre un temps `TA_EcranTesteur` (temps d'attente vers écran « Ecran_IdentificationTesteur ») pour ensuite arriver à l'écran « Ecran_IdentificationTesteur ».

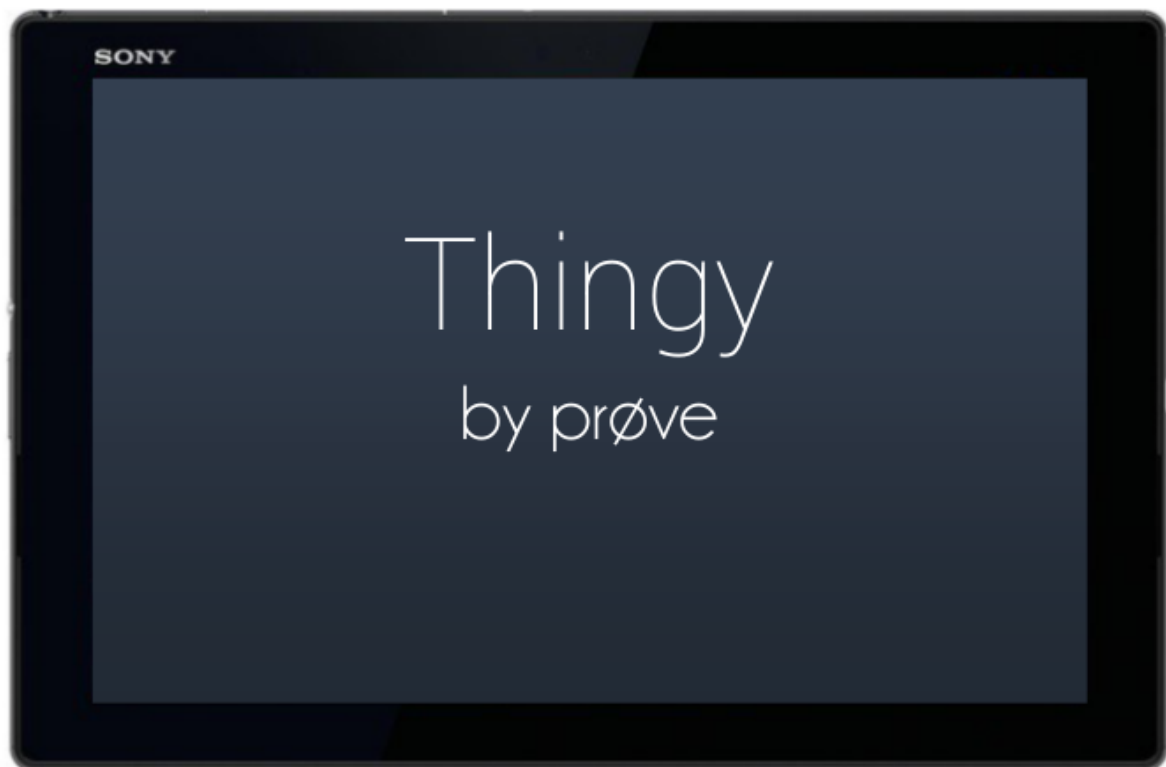


Illustration 16: IHM Ecran_Lancement

3.1.4.1.2 Dialogue_ErreurConnexion

Cette boîte de dialogue affiche un message d'erreur lorsque la communication a été interrompue. Les boutons retour de la tablette et « FERMER » permettent de fermer la boîte de dialogue.

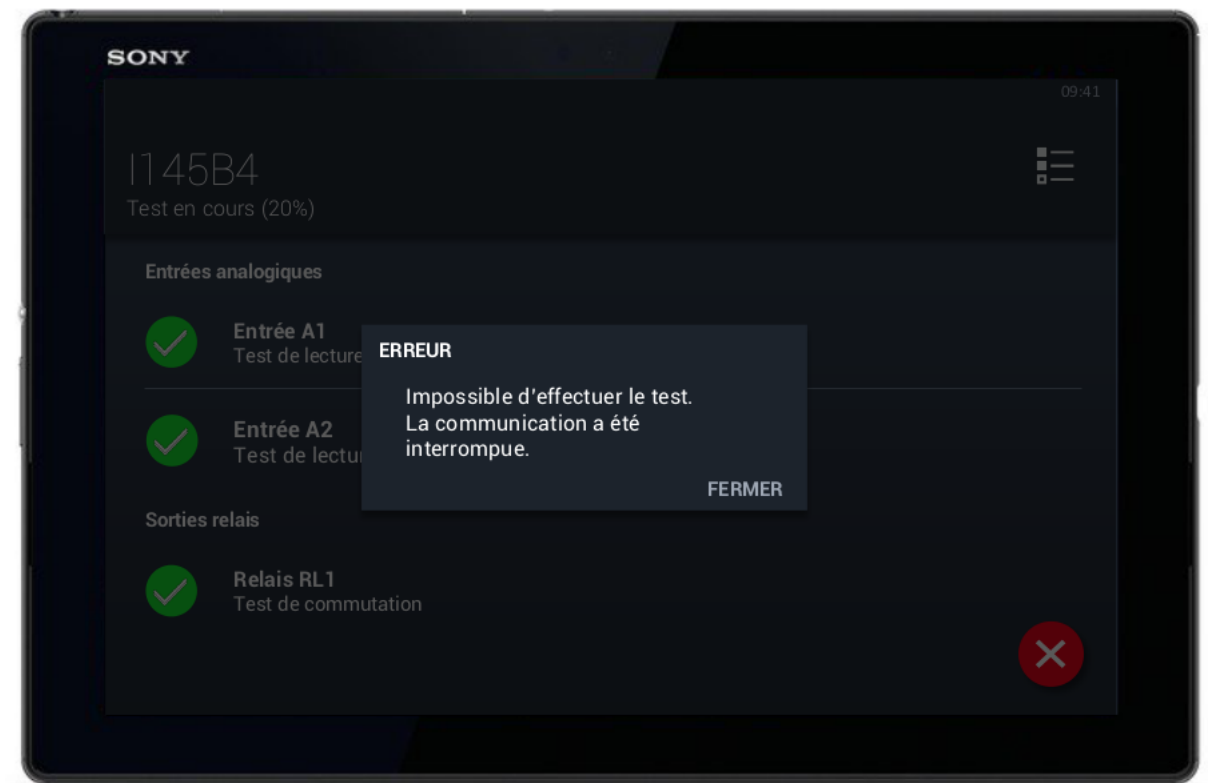


Illustration 17: IHM Dialogue_ErreurConnexion


Bouton	Intention / Action
FERMER	annuler
 Retour (bouton tablette)	annuler

Tableau 1: IHM Intention_Dialogue_ErreurConnexion

3.1.4.1.3 Dialogue_ReconnecterOuQuitter

Cette boîte de dialogue affiche un message d'erreur lorsque la communication est impossible. Le bouton quitter permet de terminer l'application et le bouton reconnecter permet de tenter une reconnexion.

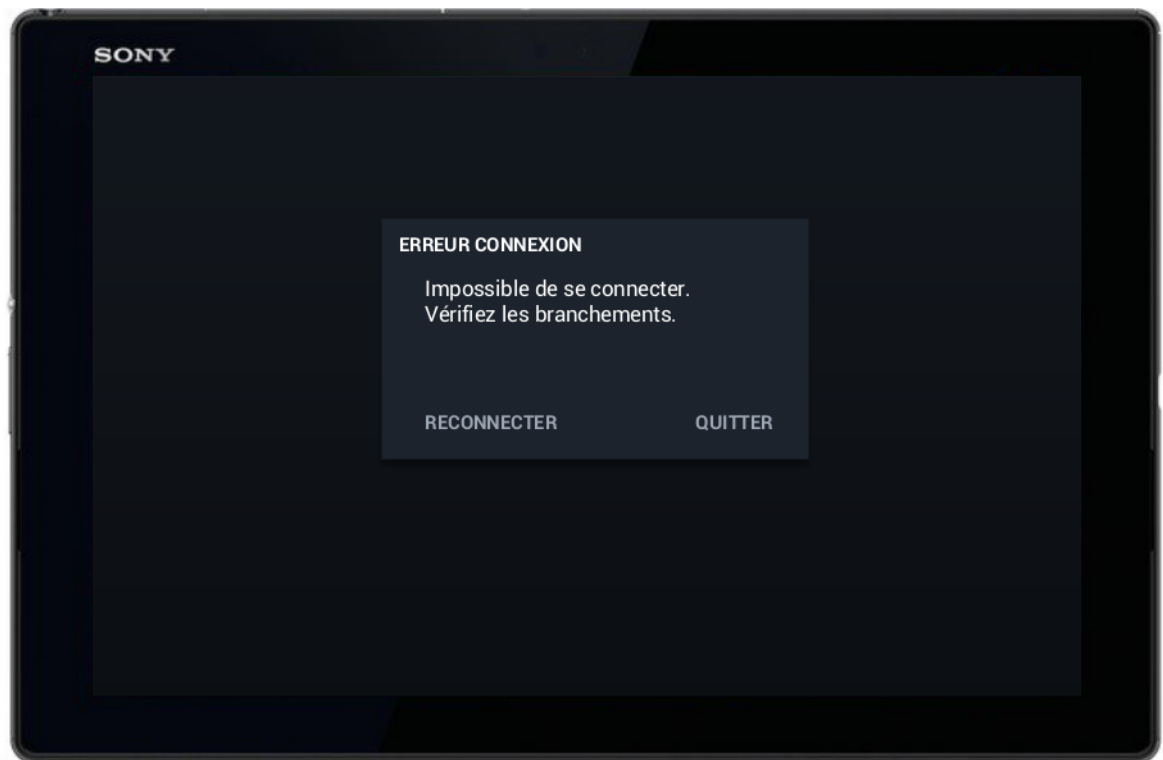


Illustration 18: IHM Dialogue_ReconnecterOuQuitter

Bouton	Intention / Action
RECONNECTER	reconnecter
QUITTER	quitter

Tableau 2: IHM Intention_Dialogue_ReconnecterOuQuitter

3.1.4.2 Depuis l'état « AppliUsing » (Illustration 14)

3.1.4.2.1 Ecran_IdentificationTesteur

Cet écran demande à l'utilisateur de saisir son identifiant et son mot de passe. Un appui sur le champ « username » permet de saisir un identifiant et le champ « password » de saisir un mot de passe. Le bouton « CONNECT » permet de valider la saisie. Le bouton retour de la tablette permet de quitter l'application.

Le champ « < message > » est la place réservée à l'IHM pour afficher des indications pour l'utilisateur.

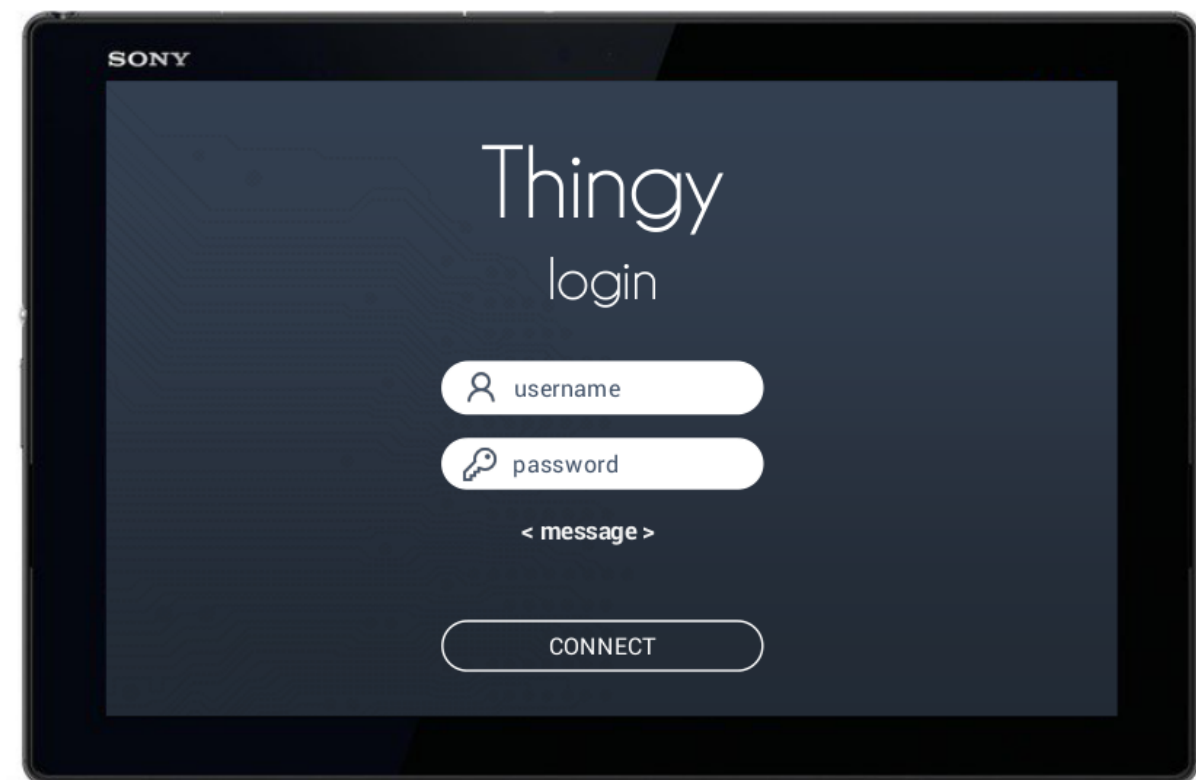


Illustration 19: IHM Ecran_IdentificationTesteur

Lors de la vérification de l'identifiant par le système, le message « connexion » s'affichera à la place prévue à cet effet (< message >).

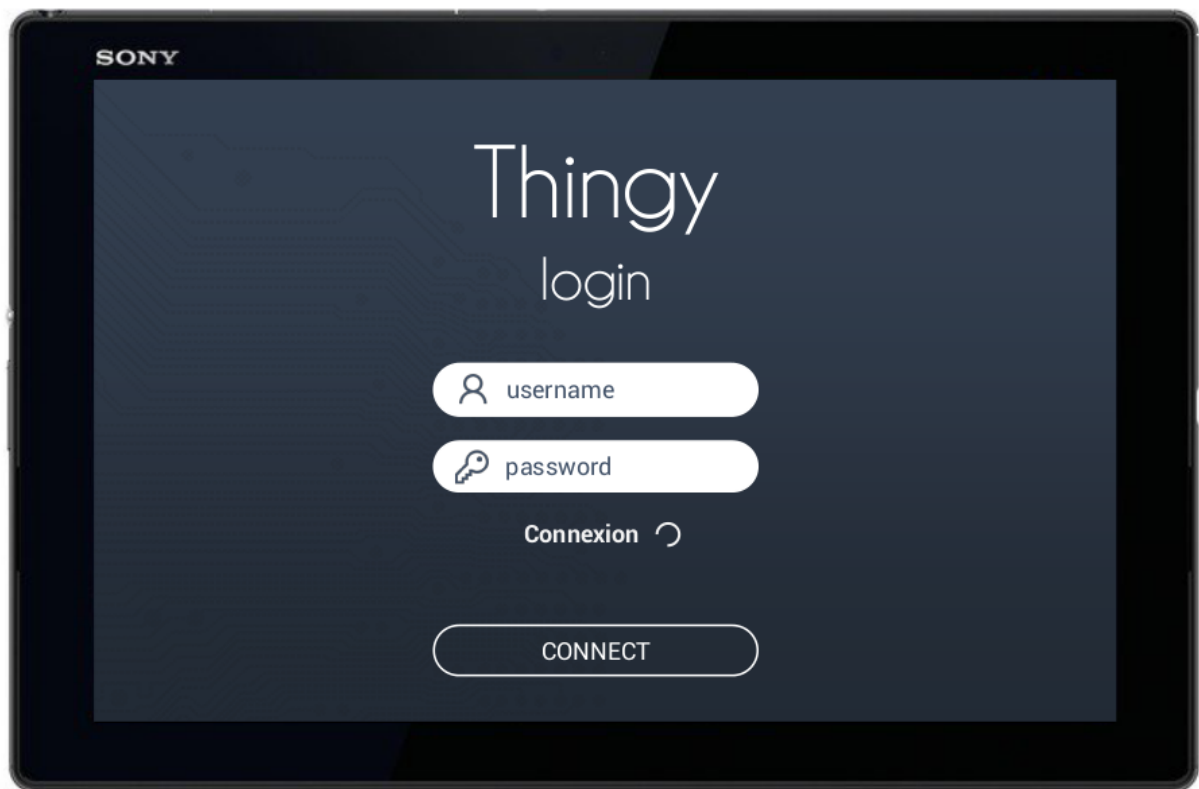


Illustration 20: IHM Message_IdentificationTesteur

Si la saisie est incorrecte, le message d'erreur « Identifiants incorrects » apparaîtra en rouge sur l'écran.

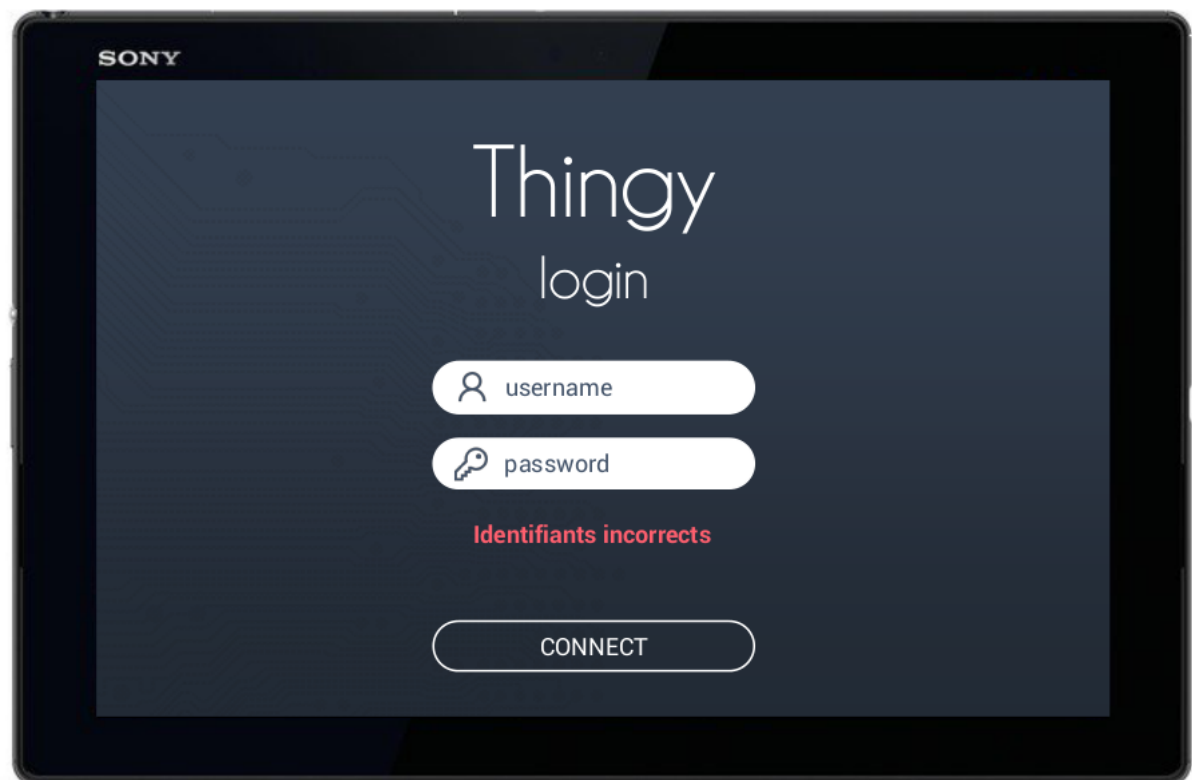


Illustration 21: IHM Erreur_IdentificationTesteur




Bouton	Intention / Action
 username	entrerIdentifiantTesteur
 password	entrerCodeTesteur
CONNECT	validerTesteur
 Retour (bouton tablette)	quitter

Tableau 3: IHM Intention_IdentificationTesteur

3.1.4.2.2 Ecran_Principal

Cet écran affiche en première ligne le nom de l'utilisateur qui s'est connecté : ici « Jean Martin », en dessous la date et l'heure de sa dernière connexion (**la date et l'heure de la dernière connexion seront implémentées dans l'incrément 3**). Dans le cadre du dessous seront affichés les derniers tests effectués sur les cartes IPX800 V4. Les paramètres affichés seront : la réussite du test « Test réussi » ou « Test échoué », la date et l'heure à laquelle ont été effectués ces tests et l'utilisateur qui les a effectués (**le temps d'exécution du test sera implémenté à l'incrément 3**).

Le bouton statistique permet d'accéder aux statistiques des tests. Le bouton représenté par une porte ou le bouton retour de la tablette permettent de changer d'utilisateur. Le bouton représenté par une loupe permet de chercher une carte pour consulter l'historique des tests appliqués sur celle-ci. Le bouton représenté par un « + » permet de chercher ou générer l'identifiant d'une carte pour lancer un test sur celle-ci. Le bouton représenté par « ESC » permet de quitter l'application.

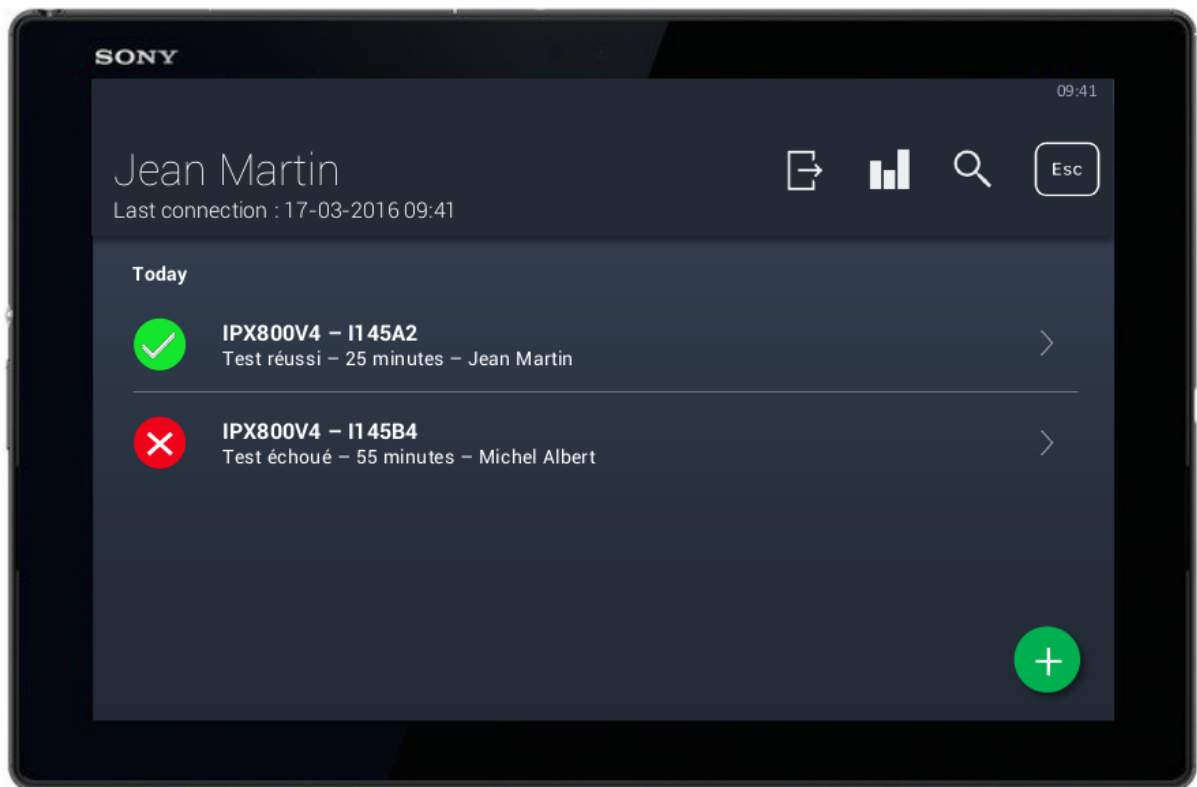


Illustration 22: IHM Ecran_Principal







Bouton	Intention / Action
	consulterHistoriqueCarte
	lancerNouveauTest
	consulterStatistiqueTest
	quitter
	deconnecter
 Retour (bouton tablette)	deconnecter

Tableau 4: IHM Intention_Ecran_Principal

3.1.4.2.3 Ecran_HistoriqueCarte

Cet écran affiche l'historique de la carte qui a été identifiée. Au départ, la carte est recherchée dans le prototype de base de données. La phrase « Recherche de la carte » sera affichée comme suit :

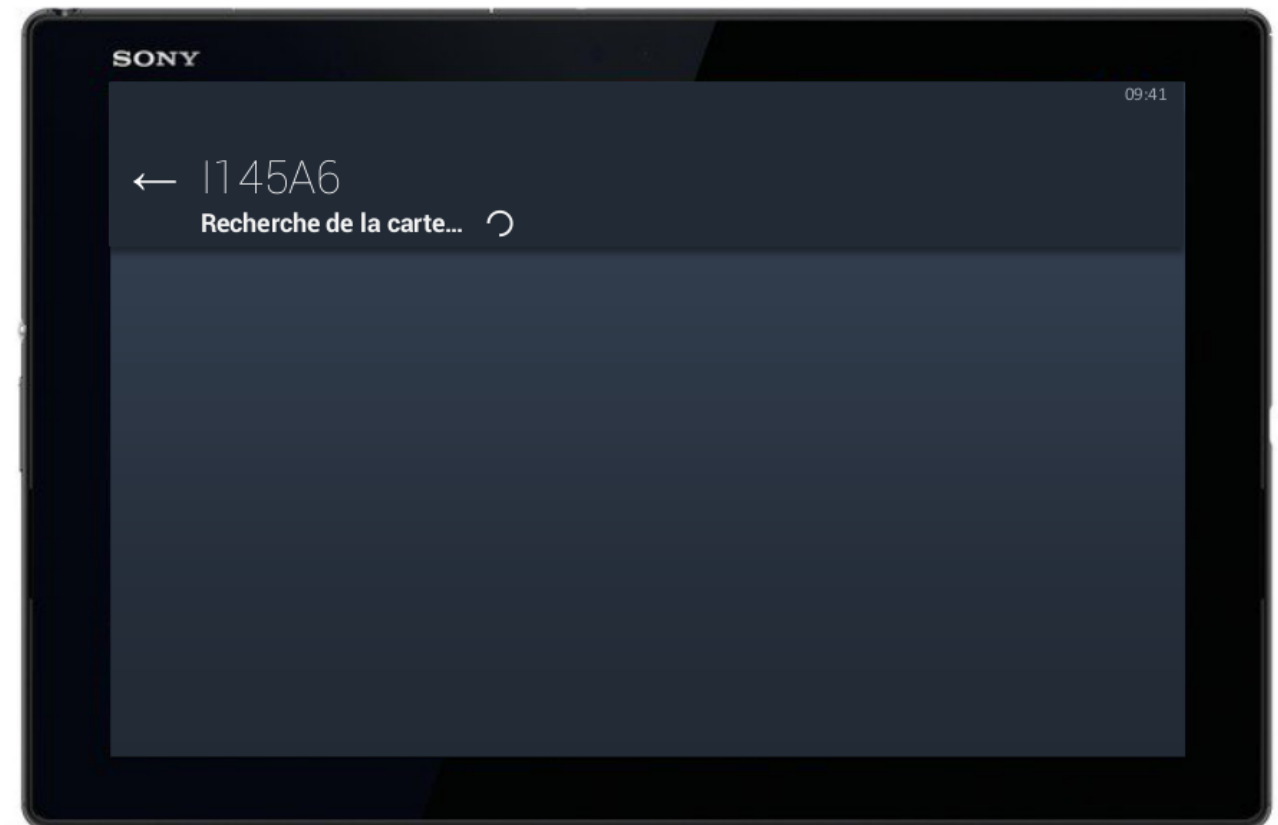


Illustration 23: IHM Ecran_HistoriqueCarte (recherche)

Si la carte n'a pas été trouvée ou que les données ne sont pas accessibles, une erreur « Carte Introuvable » s'affichera sur l'écran.

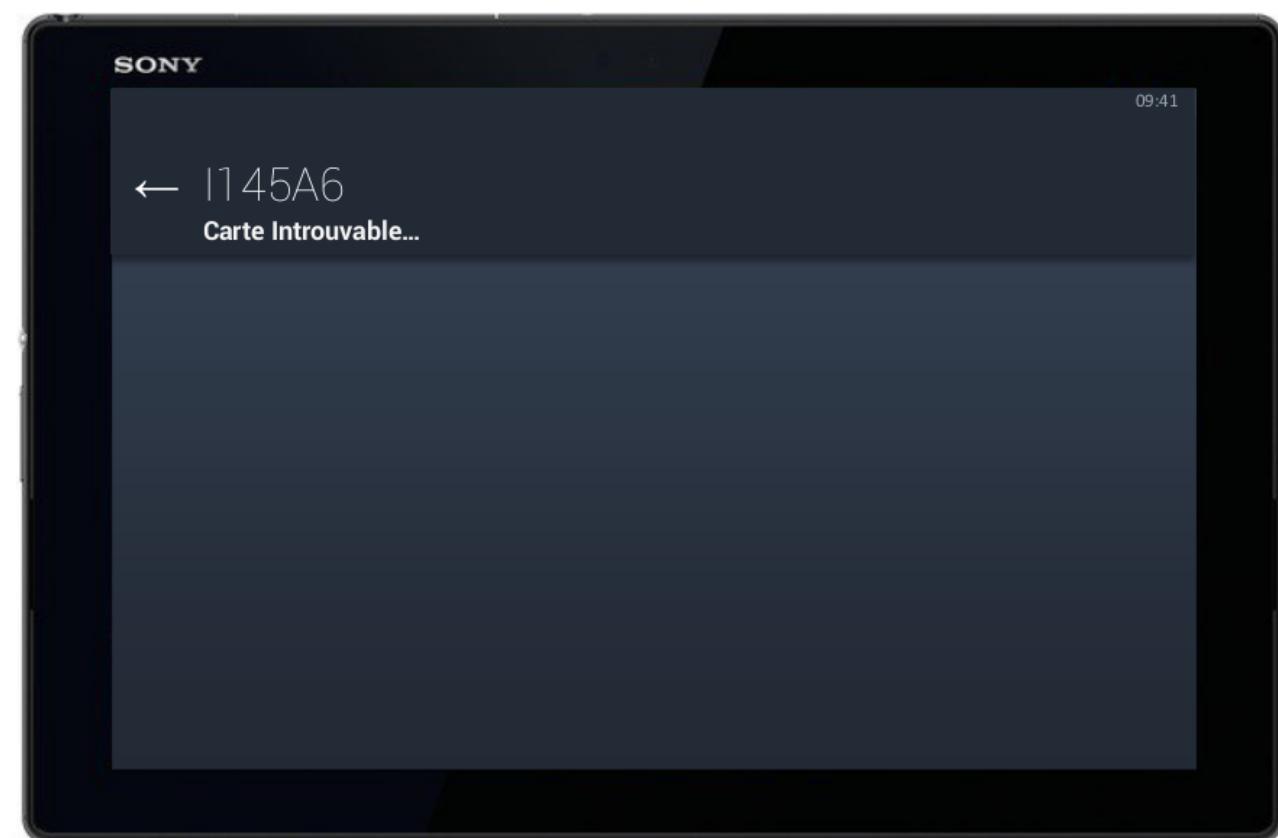


Illustration 24: IHM Ecran_HistoriqueCarte (erreur)

Si au contraire la carte a été trouvée, un écran de la sorte s'affichera :



Illustration 25: IHM Ecran_HistoriqueCarte (sans détail)

Le bouton retour de la tablette et le bouton représenté par une flèche vers la gauche permettent de revenir à l'écran « Ecran_Principal ». Le bouton représenté par cette icône « > » permet d'obtenir des détails sur le test affiché. Si l'utilisateur sélectionne le bouton détail, l'écran s'affichera comme suit :

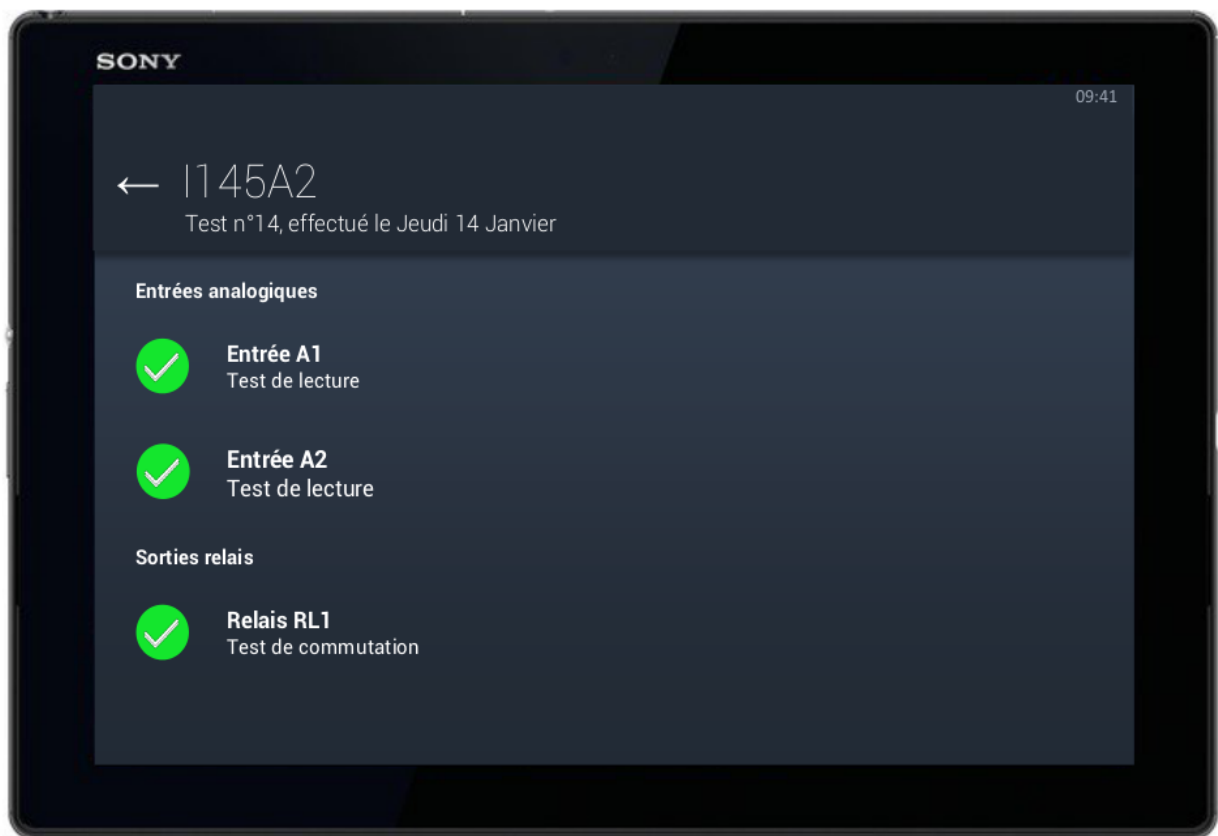


Illustration 26: IHM Ecran_HistoriqueCarte (avec détail)

Bouton	Intention / Action
←	quitter
↶ Retour (bouton tablette)	quitter
>	detail

Tableau 5: IHM Intention_Ecran_HistoriqueCarte

3.1.4.2.4 Ecran_StatistiqueTest

Cet écran affiche les statistiques des tests appliqués sur chacune des cartes IPX800 V4. Un message s'affiche au préalable sur l'écran pour accéder aux statistiques. Si les données n'ont pas pu être récupérées, un message d'erreur s'affichera (« Aucun test trouvé... »). Le bouton retour de la tablette et le bouton représenté par une flèche permettent de revenir à l'écran « Ecran_Principal ».

Cette IHM est une projection et sera implémentée durant l'incrément 3.

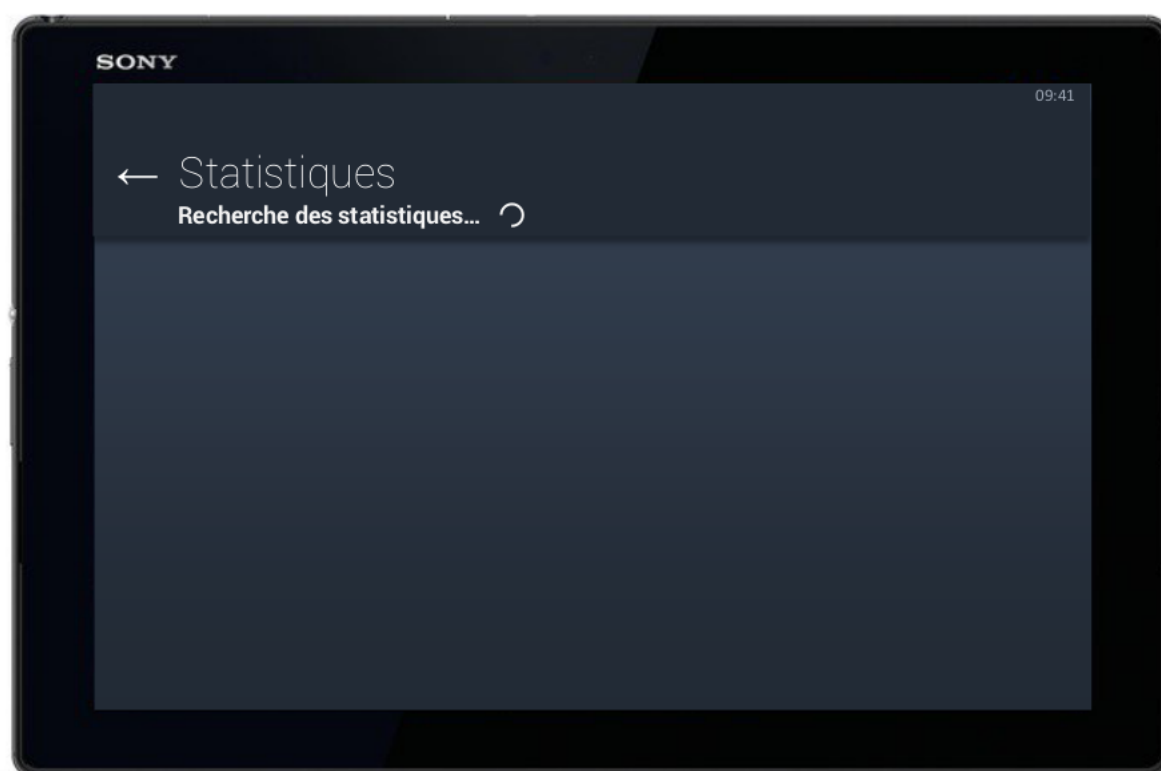


Illustration 27: IHM Ecran_StatistiqueTest (recherche)

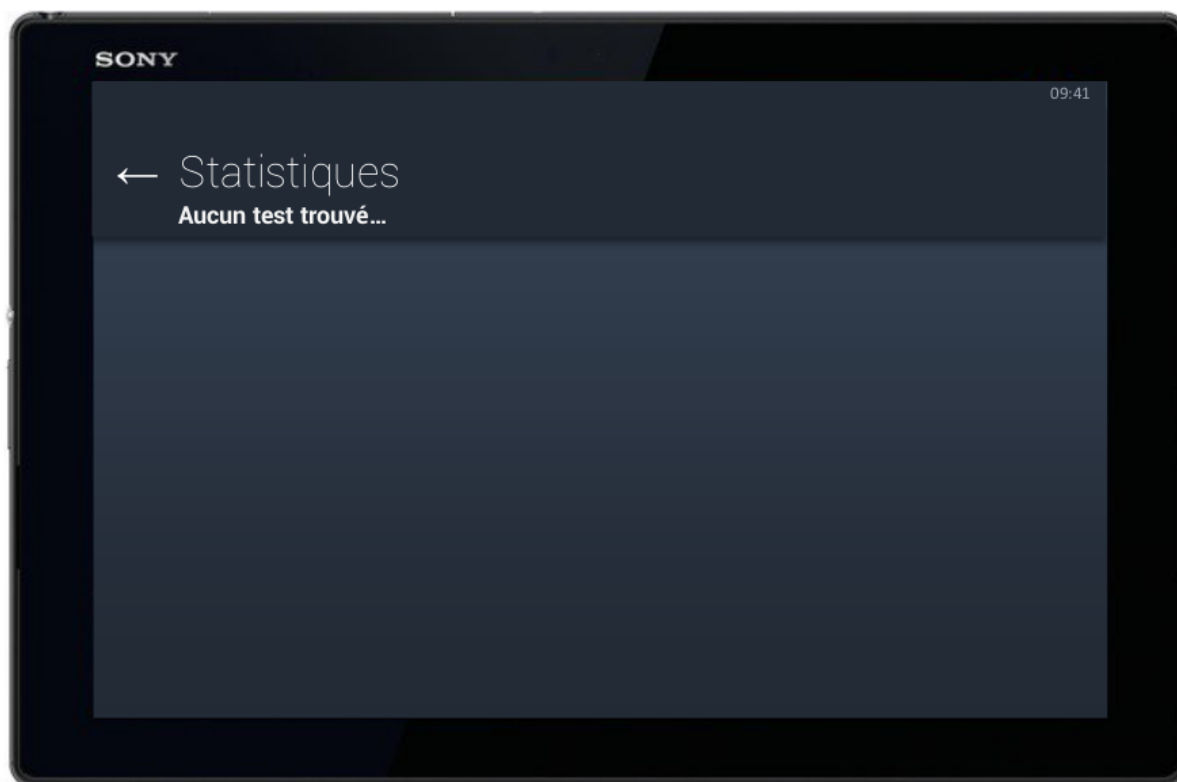


Illustration 28: IHM Ecran_StatistiqueTest (erreur)



Illustration 29: IHM Ecran_StatistiqueTest (avec statistiques)

Bouton	Intention / Action
←	quitter
↶ Retour (bouton tablette)	quitter

Tableau 6: IHM Intention_Ecran_StatistiqueTest

3.1.4.2.5 Dialogue_TesteurDeconnexion

Cette boîte de dialogue affiche un message demandant à l'utilisateur s'il est sûr de vouloir déconnecter l'utilisateur en cours. Les boutons retour de la tablette et « ANNULER » permettent de fermer la boîte de dialogue et le bouton « CONFIRMER » permet de confirmer la déconnexion.



Illustration 30: IHM Dialogue_TesteurDeconnexion


Bouton	Intention / Action
CONFIRMER	valider
ANNULER	annuler
 Retour (bouton tablette)	annuler

Tableau 7: IHM Intention_Dialogue_TesteurDeconnexion

3.1.4.2.6 Dialogue_RechercheCarte

Cette boîte de dialogue affiche un message demandant à l'utilisateur de rentrer une carte pour laquelle il souhaite consulter l'historique. En appuyant sur le champ « < EX : XXJJMMAAHMM > », le clavier virtuel s'affiche pour permettre à l'utilisateur de saisir l'identifiant de la carte. Le bouton « VALIDER » permet de valider la saisie de l'identifiant de la carte. Les boutons retour de la tablette et « ANNULER » permettent de fermer la boîte de dialogue.

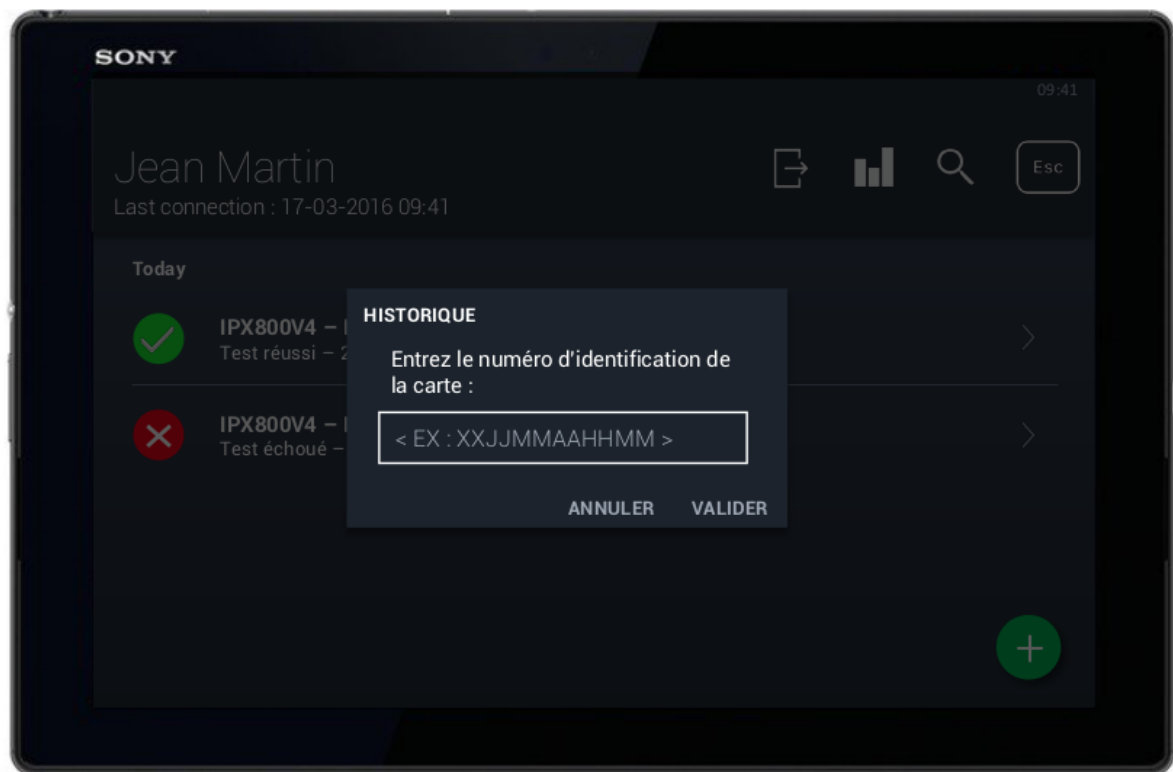


Illustration 31: IHM Dialogue_RechercheCarte


Bouton	Intention / Action
< EX : XXJJMMAAHMM >	saisirIdentifiantCarte
VALIDER	valider
ANNULER	annuler
 Retour (bouton tablette)	annuler

Tableau 8: IHM Intention_Dialogue_RechercheCarte

3.1.4.2.7 Dialogue_CreationRecherche

Cette boîte de dialogue fait comprendre à l'utilisateur qu'une recherche ou une création d'identifiant de carte est en cours. Cette boîte de dialogue se ferme dès que le système aura créé un identifiant, trouvé la carte identifiée ou levé une erreur si la carte n'a pas été trouvée.

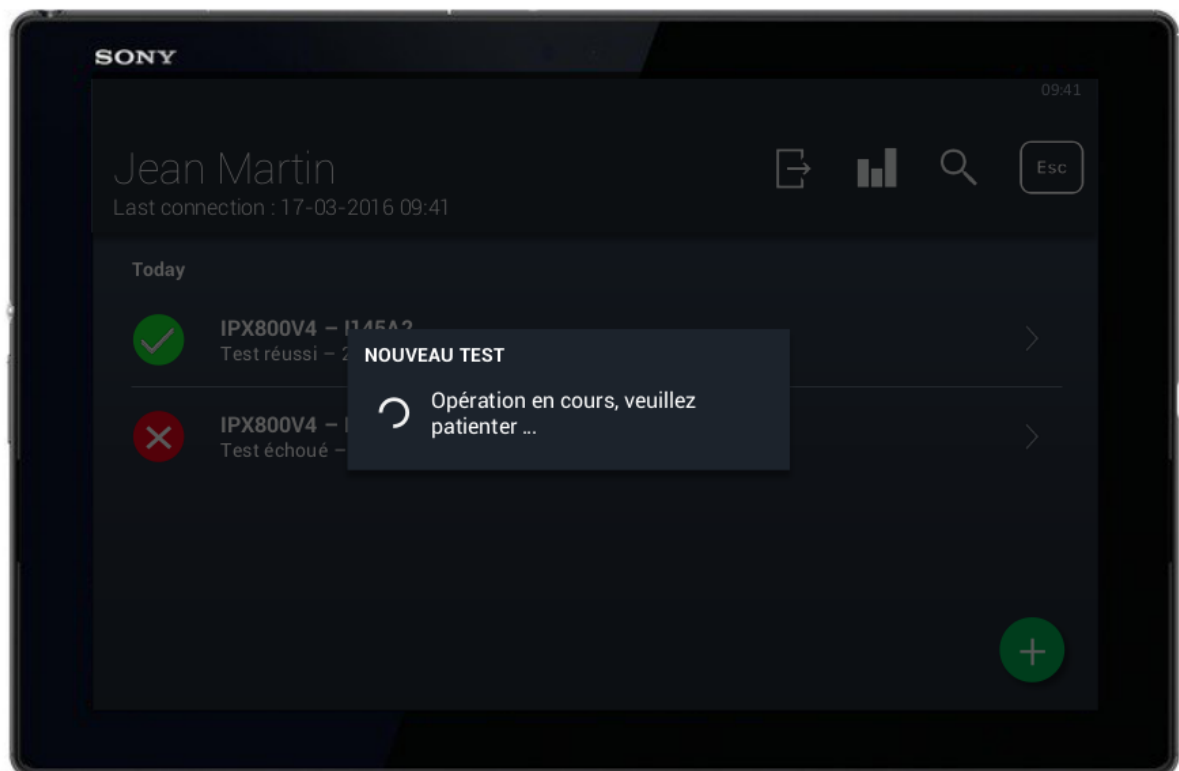


Illustration 32: IHM Dialogue_CreationRecherche

3.1.4.2.8 Dialogue_ErreurIdCarte

Cette boîte de dialogue affiche un message d'erreur lorsque la carte identifiée n'a pas été trouvée. Les boutons retour de la tablette et « FERMER » permettent de fermer la boîte de dialogue.

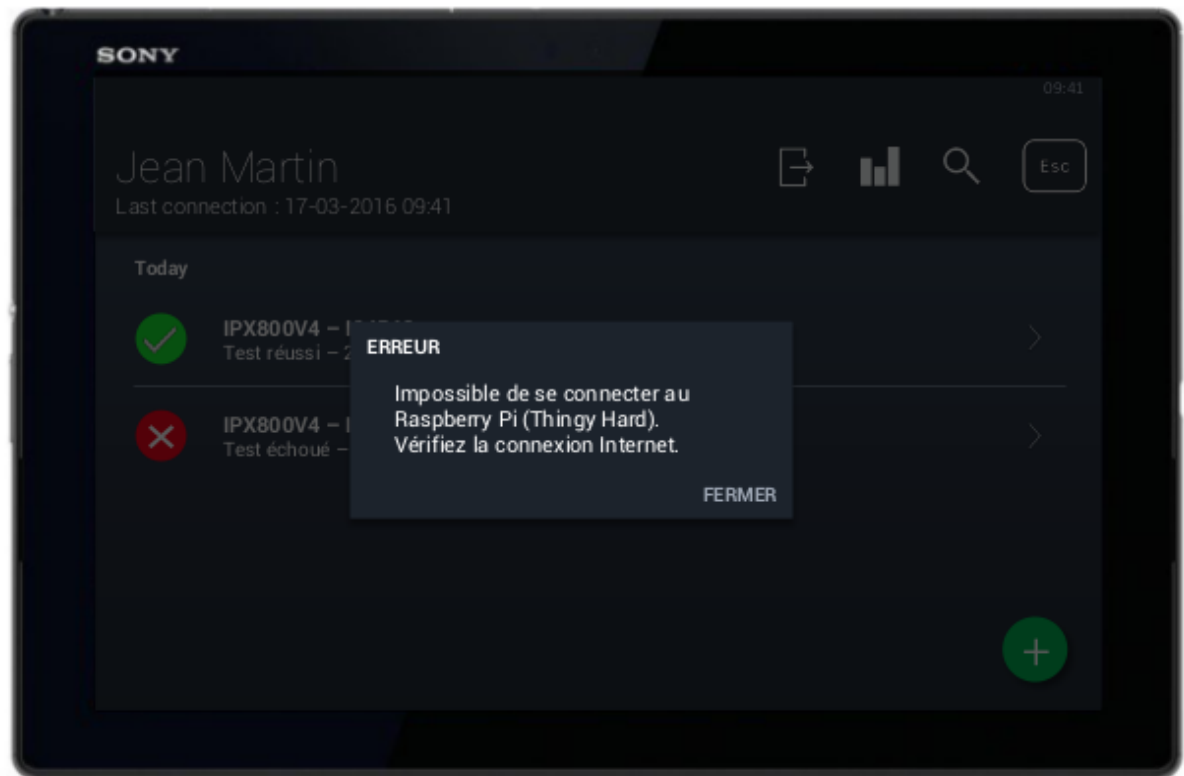


Illustration 33: IHM Dialogue_ErreurIdCarte

Bouton	Intention / Action
FERMER	annuler
Retour (bouton tablette)	annuler

Tableau 9: IHM Intention_Dialogue_ErreurIdCarte

3.1.4.2.9 Dialogue_ConfirmerQuitter

Cette boîte de dialogue affiche un message demandant à l'utilisateur s'il est sûr de vouloir quitter. Le bouton « CONFIRMER » confirme, les boutons retour de la tablette et « ANNULER » permettent de fermer la boîte de dialogue et de revenir à l'écran précédent.

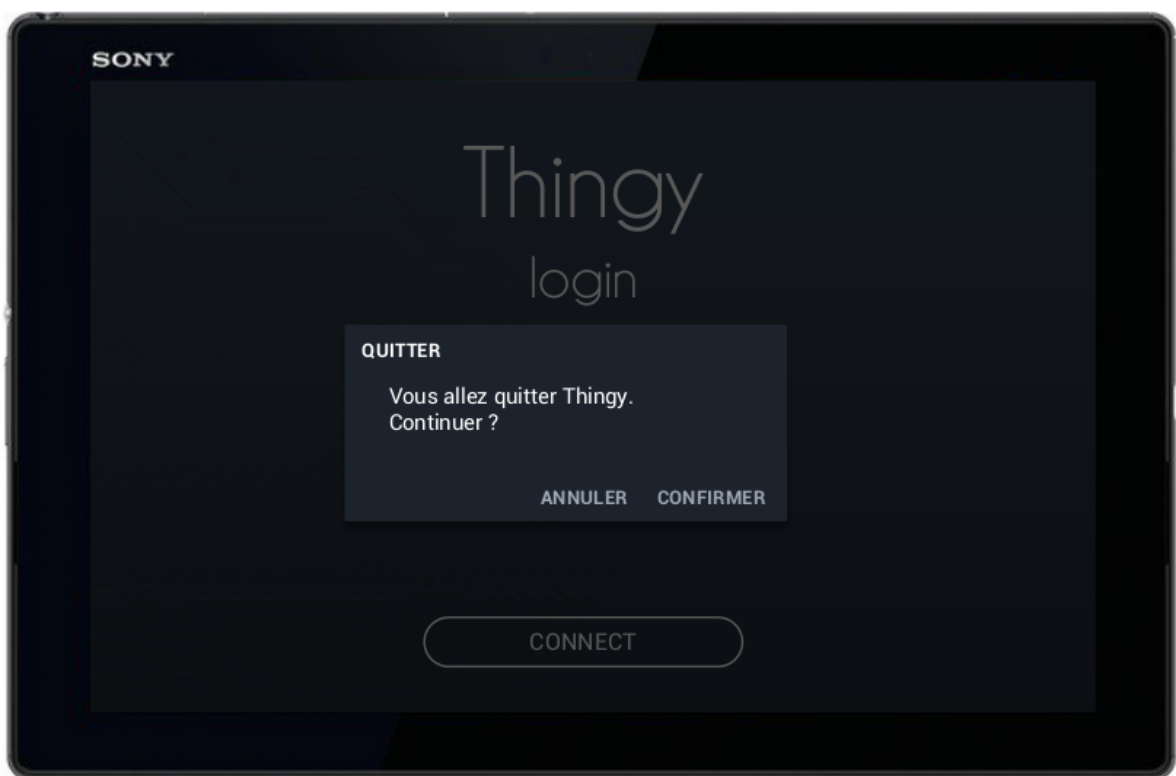


Illustration 34: IHM Dialogue_ConfirmerQuitter1 (depuis login)

Depuis le menu home, ce dialogue s'affiche de la manière suivante :

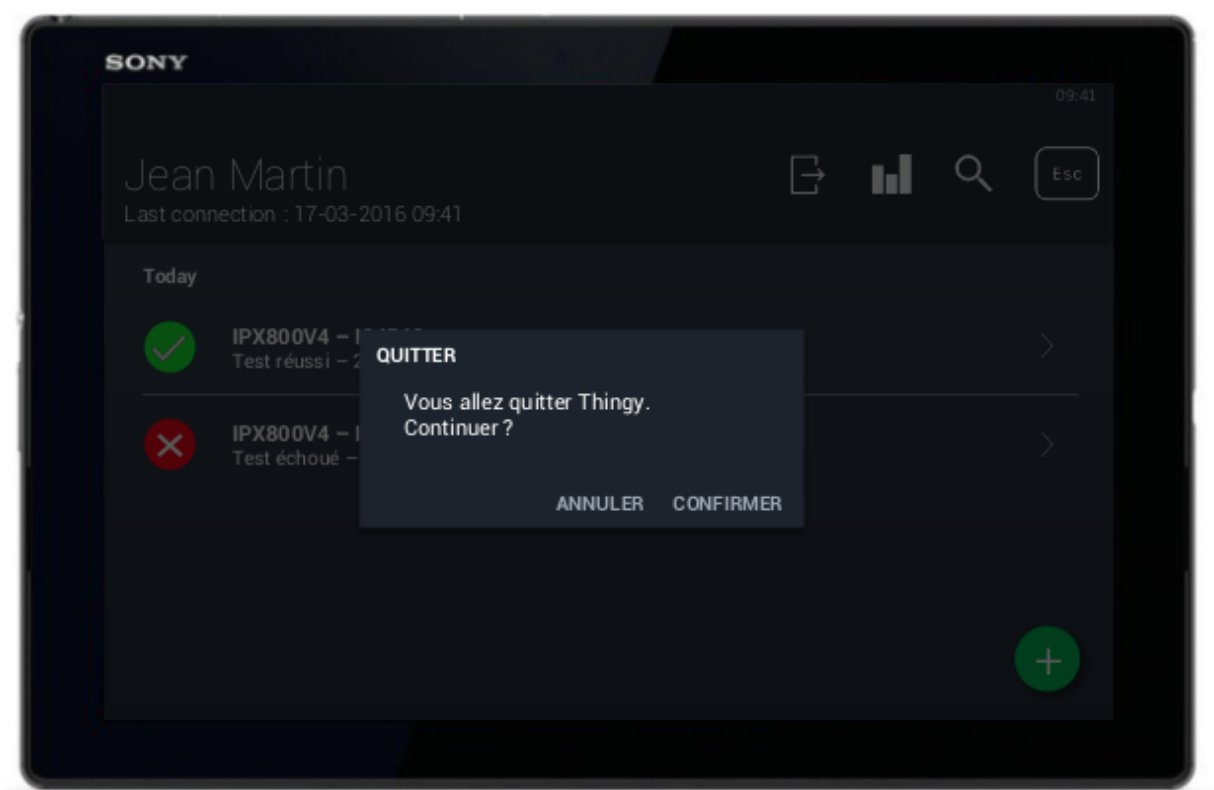


Illustration 35: IHM Dialogue_ConfirmerQuitter2 (depuis home)


Bouton	Intention / Action
CONFIRMER	valider
ANNULER	annuler
 Retour (bouton tablette)	annuler

Tableau 10: IHM Intention_Dialogue_ConfirmerQuitter

3.1.4.3 Depuis « Dialogues IdentificationCarte »

3.1.4.3.1 Dialogue_IdNewCarte

Cette boîte de dialogue affiche un message demandant à l'utilisateur s'il souhaite créer une nouvelle carte. Le bouton « VALIDER » valide la génération d'un nouvel identifiant. Décocher la case « Créer une nouvelle carte » permet d'afficher la boîte de dialogue « Dialogue_IdCarte ». Les boutons retour de la tablette et « ANNULER » permettent de fermer cette boîte de dialogue.

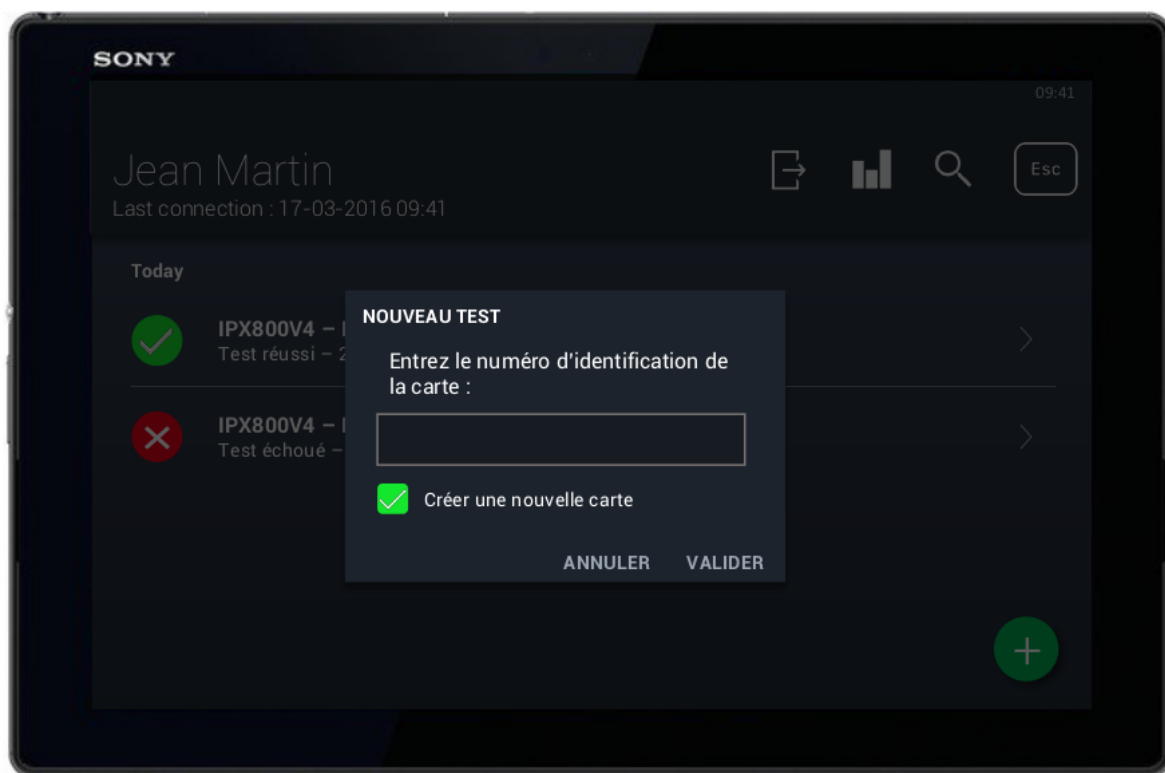


Illustration 36: IHM Dialogue_IdNewCarte


Bouton	Intention / Action
	decocher
VALIDER	valider
ANNULER	annuler
Retour (bouton tablette)	annuler

Tableau 11: IHM Intention_Dialogue_IdNewCarte

3.1.4.3.2 Dialogue_IdCarte

Cette boîte de dialogue affiche un message demandant à l'utilisateur de rentrer l'identifiant de la carte à tester. Un appui sur le champ « Ex : ABC123 » affichera un clavier virtuel permettant de saisir un identifiant. Le bouton « VALIDER » permet de valider la saisie de l'identifiant. Cocher la case « Créer une nouvelle carte » permet d'afficher la boîte de dialogue « Dialogue_IdNewCarte ». Les bouton retour de la tablette ou « ANNULER » permettent de fermer cette boîte de dialogue.

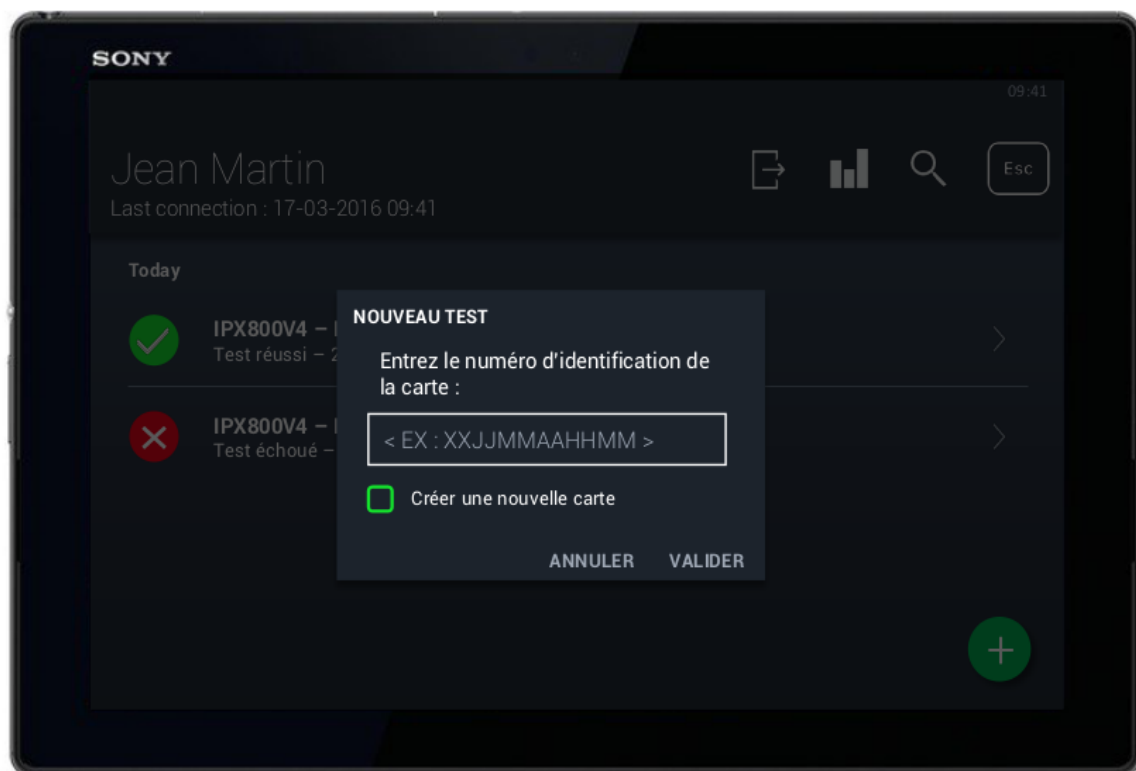


Illustration 37: IHM Dialogue_IdCarte



Bouton	Intention / Action
	cocher
<input type="text" value=" < EX : XXJJMMAAHMM >"/>	saisirIdentifiantCarte
VALIDER	valider
ANNULER	annuler
 Retour (bouton tablette)	annuler

Tableau 12: IHM Intention_Dialogue_IdCarte

3.1.4.4 Depuis « Ecrans Test »

3.1.4.4.1 Ecran_AttenteTest

Cet écran affiche les tests qui vont être réalisés. Le bouton représenté par un triangle blanc sur fond vert permet de démarrer le(s) test(s). Le bouton retour de la tablette permet de quitter l'application. Le bouton d'édition des tests permet de configurer les tests.

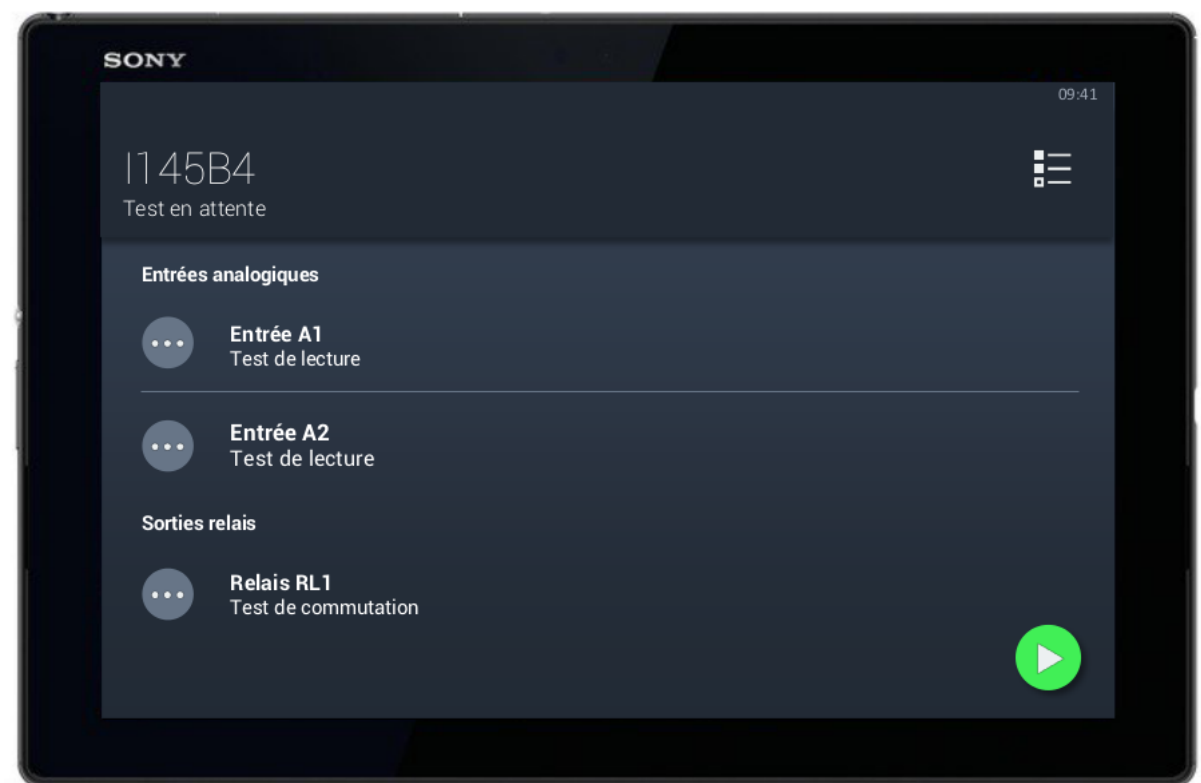


Illustration 38: IHM Ecran_AttenteTest




Bouton	Intention / Action
	demarrerTest
	editerTest
 Retour (bouton tablette)	quitter

Tableau 13: IHM Intention_Ecran_AttenteTest

3.1.4.4.2 Ecran_EditionTest

Cet écran affiche les tests qui peuvent être configurés. Sur cet écran, décocher un test permet de ne pas réaliser un test. Cocher un test permet de le réaliser. Le bouton valider permet de valider la configuration. Le bouton retour de la tablette ou annuler permettent de ne pas valider la configuration.

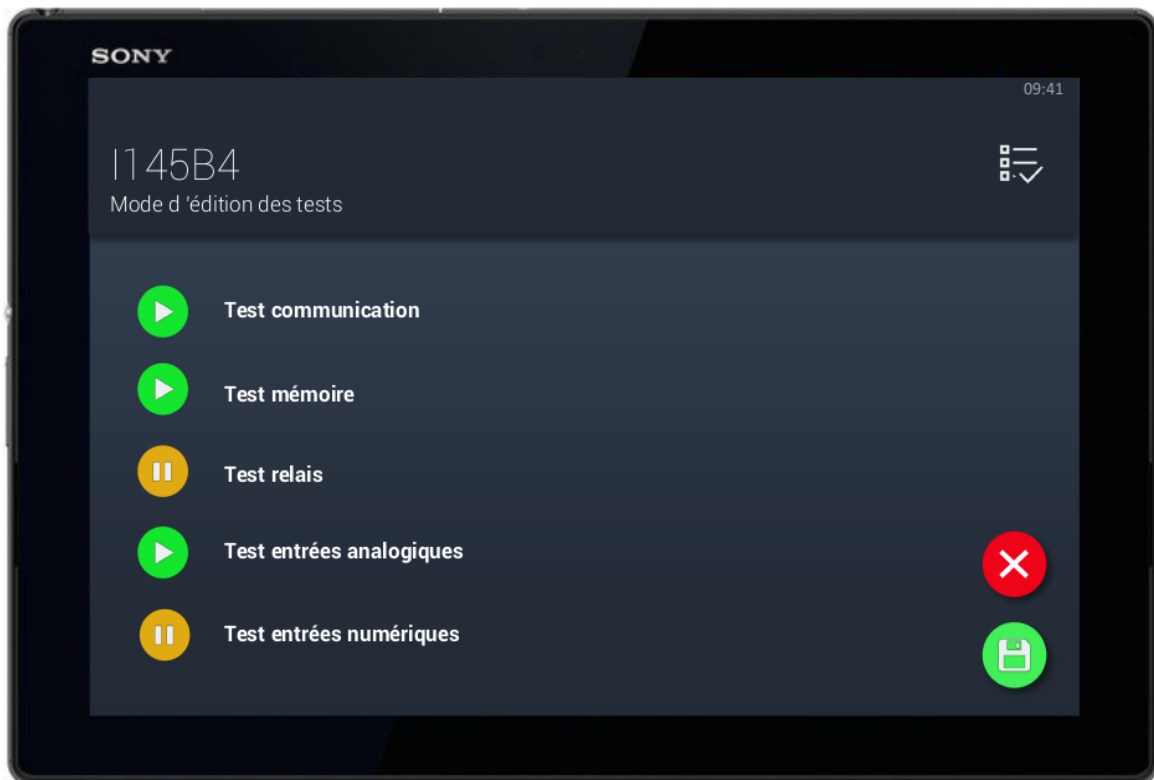


Illustration 39: IHM Ecran_EditionTest (avant sauvegarde)

Si l'utilisateur valide la configuration, l'icône de validation en bas à droite se modifiera comme suit pour indiquer à l'utilisateur que la sauvegarde de la configuration est en cours.

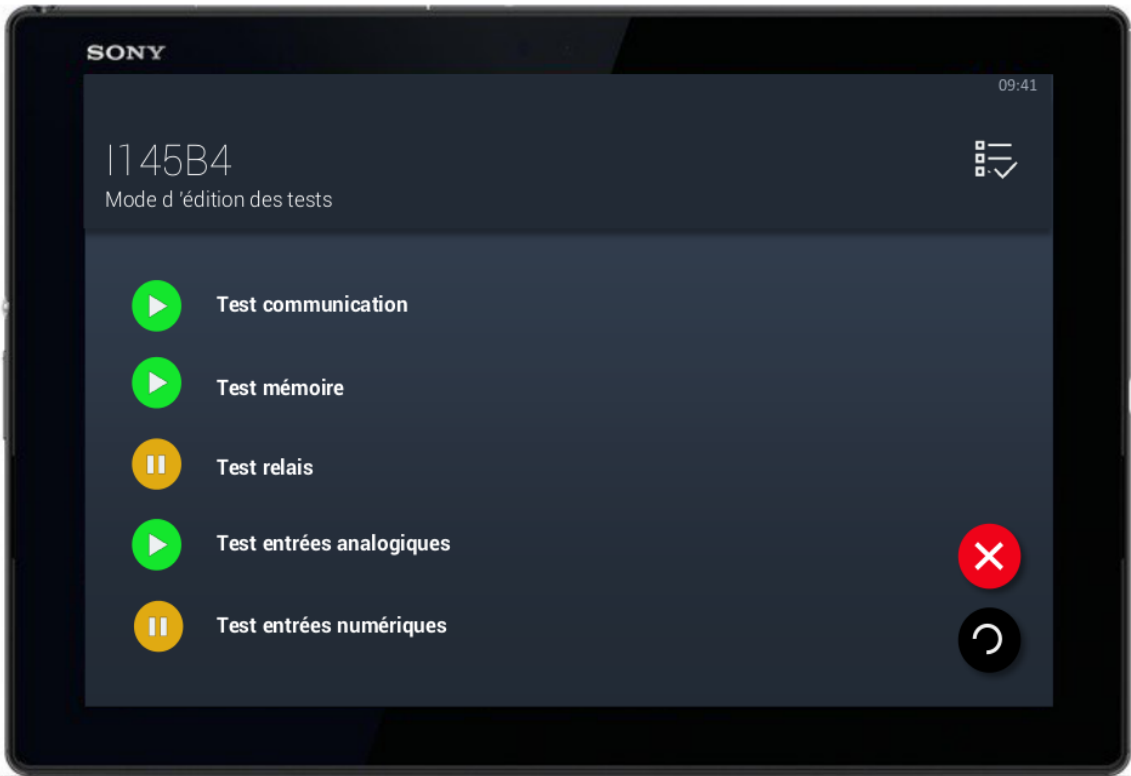


Illustration 40: IHM Ecran_EditionTest (pendant sauvegarde)







Bouton	Intention / Action
	validerEdition
	validerEdition
	cocher
	decocher
	annuler
 Retour (bouton tablette)	annuler

Tableau 14: IHM Intention_Ecran_EditionTest

3.1.4.4.3 Ecran_ProgressionTest

Cet écran affiche la progression du test qui a été lancé. L'utilisateur devra attendre la fin des tests pour pouvoir voir les résultats. Le bouton retour de la tablette ou annuler permettent d'annuler les tests en cours.

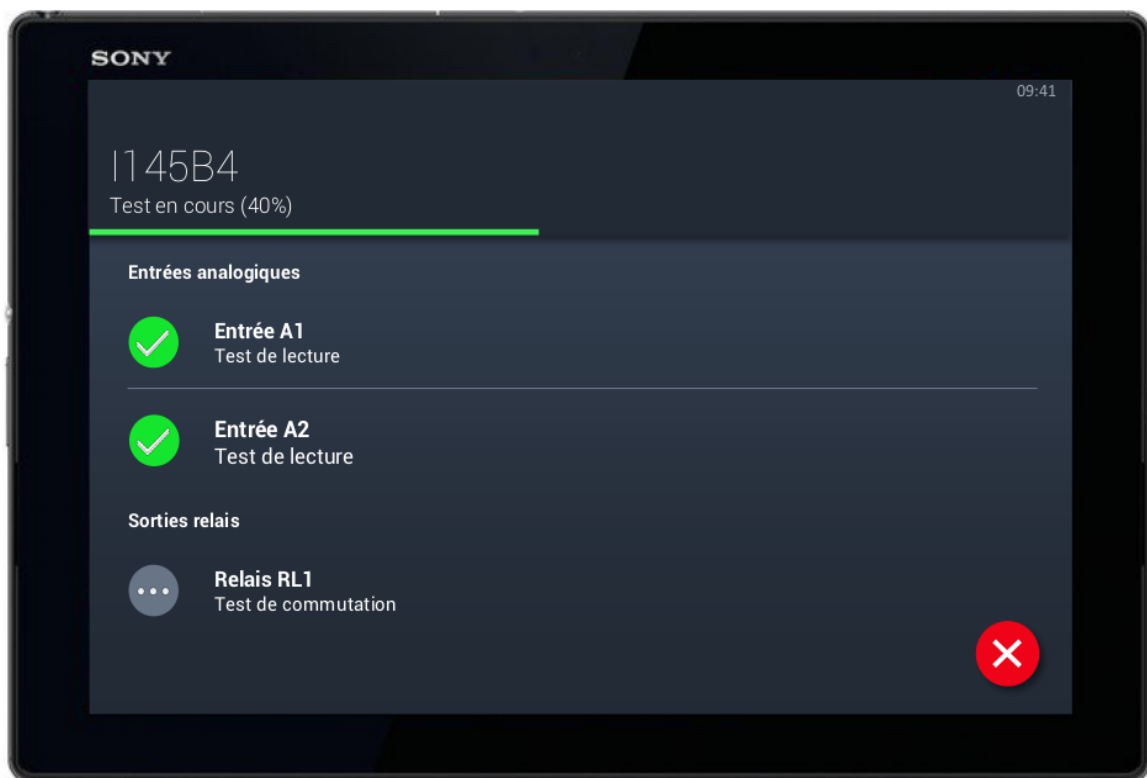


Illustration 41: IHM Ecran_ProgressionTest



Bouton	Intention / Action
	annuler
 Retour (bouton tablette)	annuler

Tableau 15: IHM Intention_Ecran_ProgressionTest

3.1.4.4.4 Ecran_ResultatTest

Cet écran affiche les résultats des tests qui viennent d'être exécutés. Les deux boutons en bas à droite de l'écran permettent de valider ou d'invalider les tests exécutés. Le bouton retour de la tablette ou celui représenté par une porte permettent de quitter l'application. Le bouton correspondant à rejouer les tests permet de refaire des tests sur la carte.

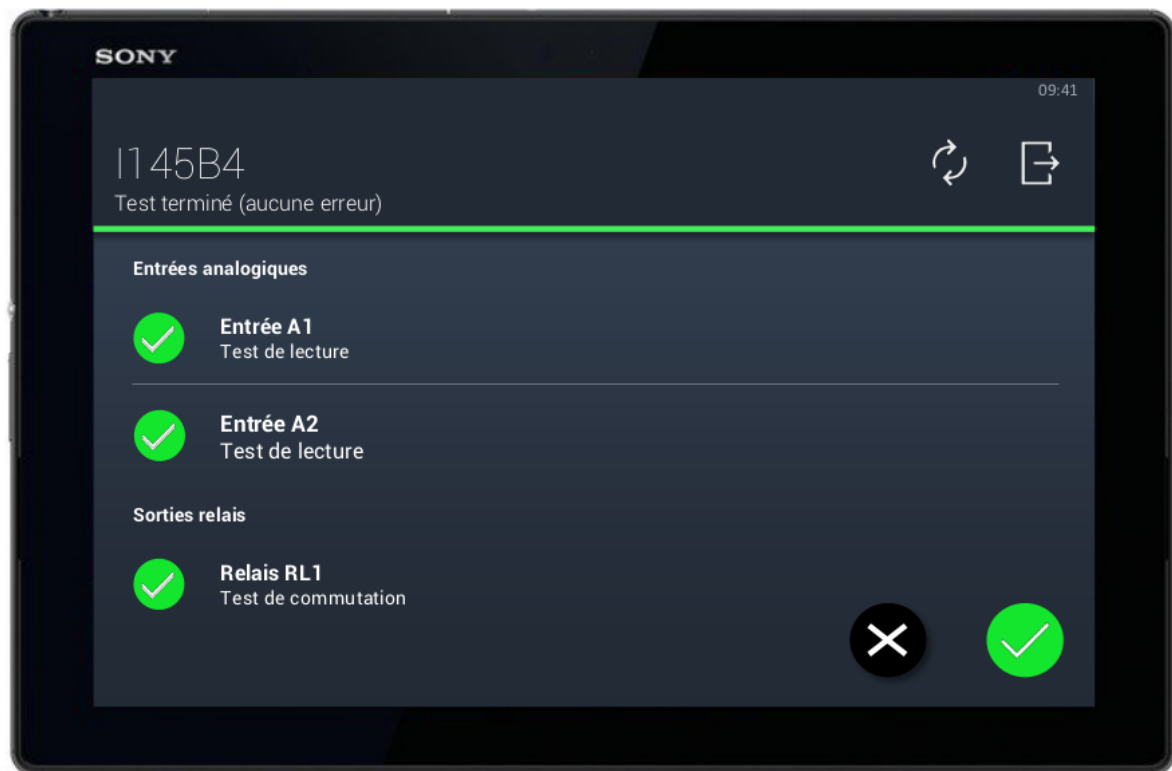


Illustration 42: IHM Ecran_ResultatTest (avant intention)

Si l'utilisateur décide de rejouer les tests, l'icône pour rejouer les tests se modifiera comme suit pour indiquer à l'utilisateur que la sauvegarde des résultats des tests est en cours.

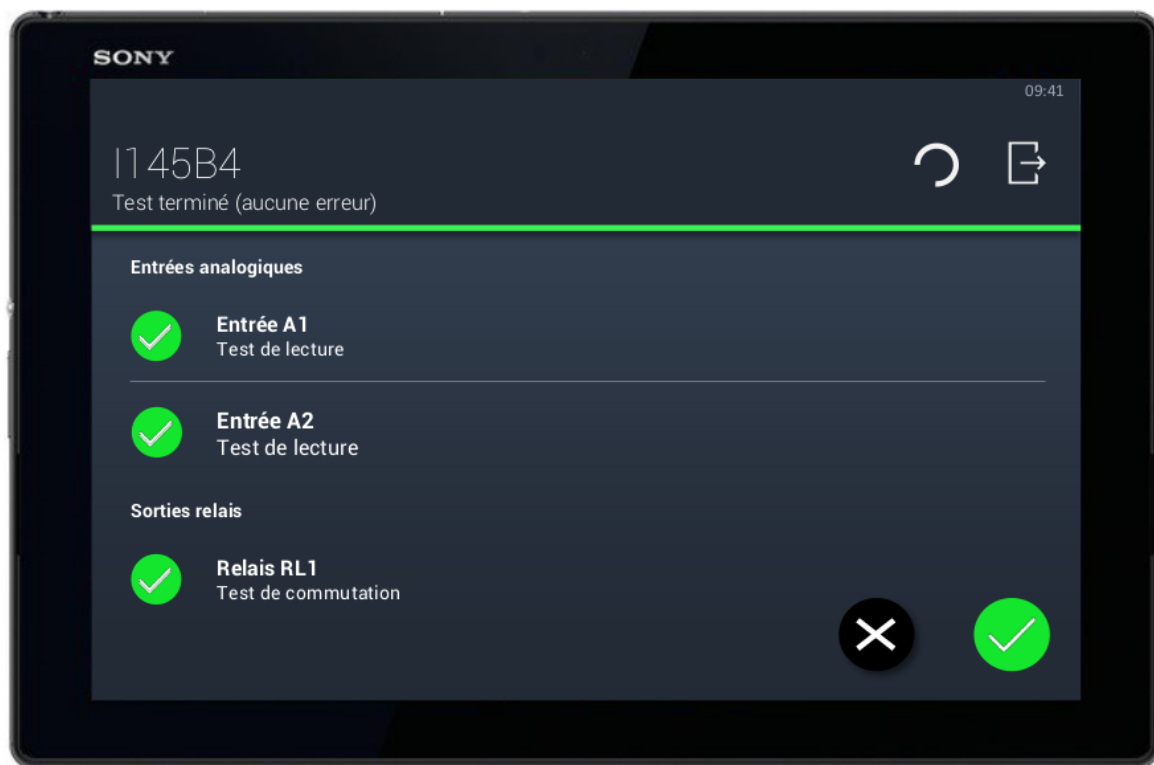


Illustration 43: IHM Ecran_ResultatTest (pendant sauvegarde)






Bouton	Intention / Action
	validerResultat
	invaliderResultat
	rejouerTest
	quitter
 Retour (bouton tablette)	quitter

Tableau 16: IHM Intention_Ecran_ResultatTest

3.1.4.4.5 Dialogue_ConfirmerQuitter

Cette boîte de dialogue affiche un message demandant à l'utilisateur s'il est sûr de vouloir quitter. Le bouton « CONFIRMER » permet de quitter et le bouton retour de la tablette ou « ANNULER » permettent de fermer la boîte de dialogue.

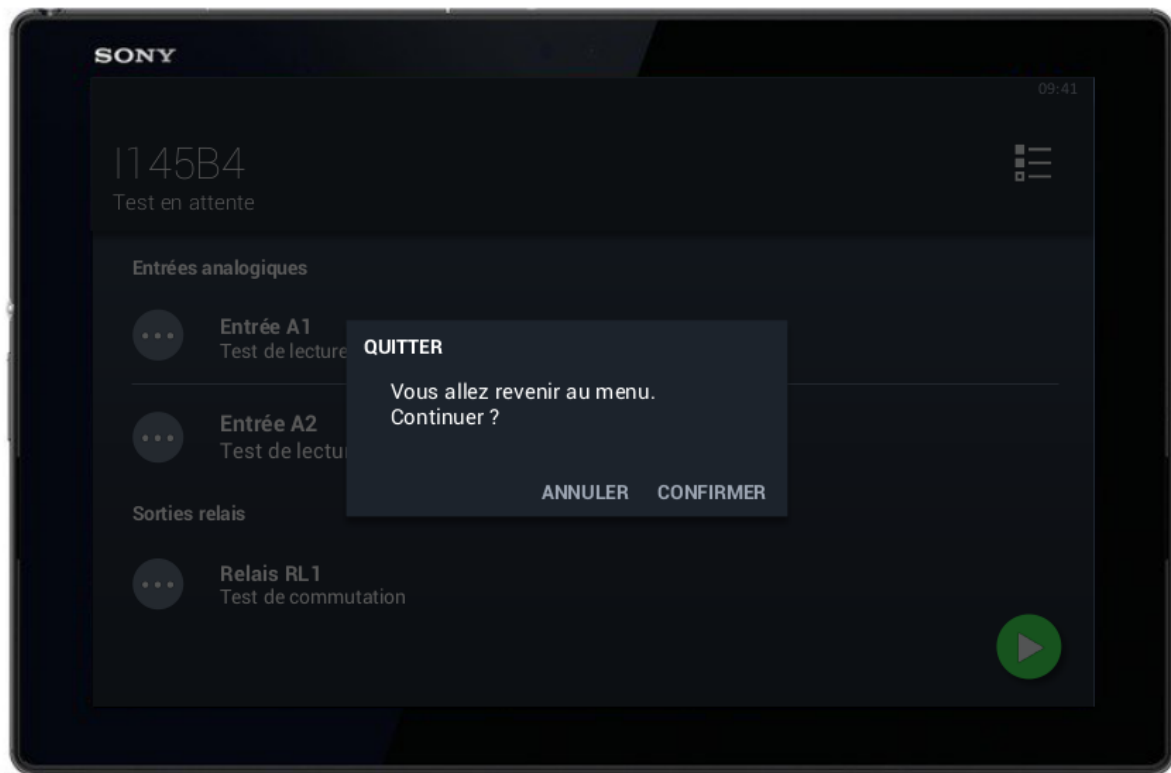


Illustration 44: IHM Dialogue_ConfirmerQuitter2 (depuis attente test)

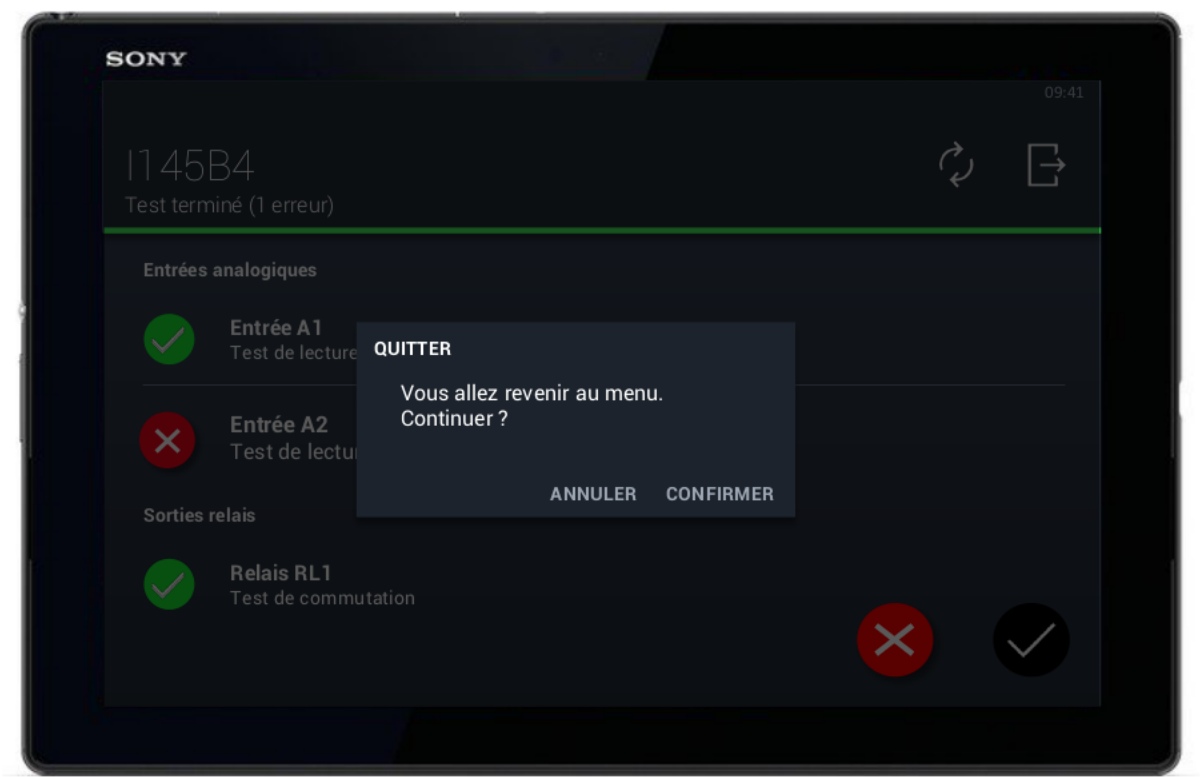


Illustration 45: IHM Dialogue_ConfirmerQuitter2 (depuis résultat test)


Bouton	Intention / Action
CONFIRMER	valider
ANNULER	annuler
 Retour (bouton tablette)	annuler

Tableau 17: IHM Intention_Dialogue_ConfirmerQuitter2

Si l'utilisateur valide, la boîte de dialogue se modifiera comme suit pour indiquer à l'utilisateur que la sauvegarde des résultats des tests est en cours.

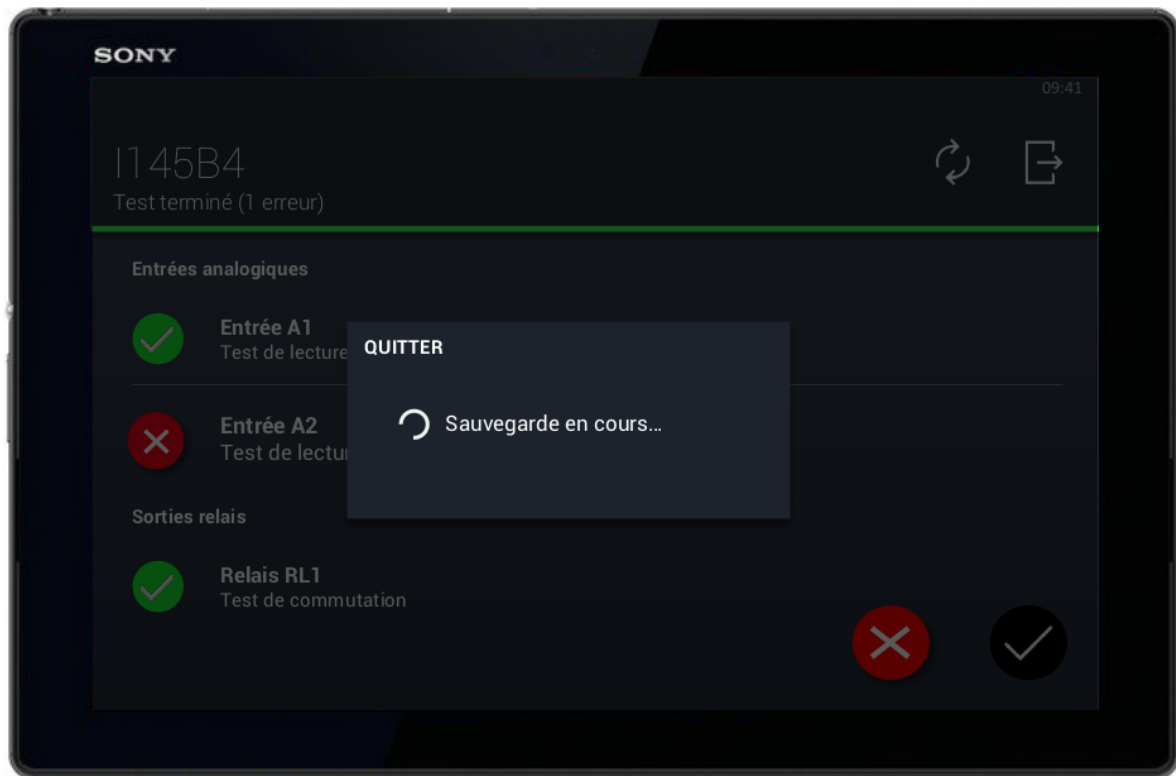


Illustration 46: IHM Dialogue_ConfirmerQuitter3 (pendant sauvegarde)

3.1.4.4.6 Dialogue_ConfirmerInterruption

Cette boîte de dialogue affiche un message demandant à l'utilisateur s'il est sûr de vouloir interrompre le(s) test(s) en cours. Le bouton « CONFIRMER » valide et le bouton retour de la tablette ou « ANNULER » permettent de fermer la boîte de dialogue et de reprendre le(s) test(s).

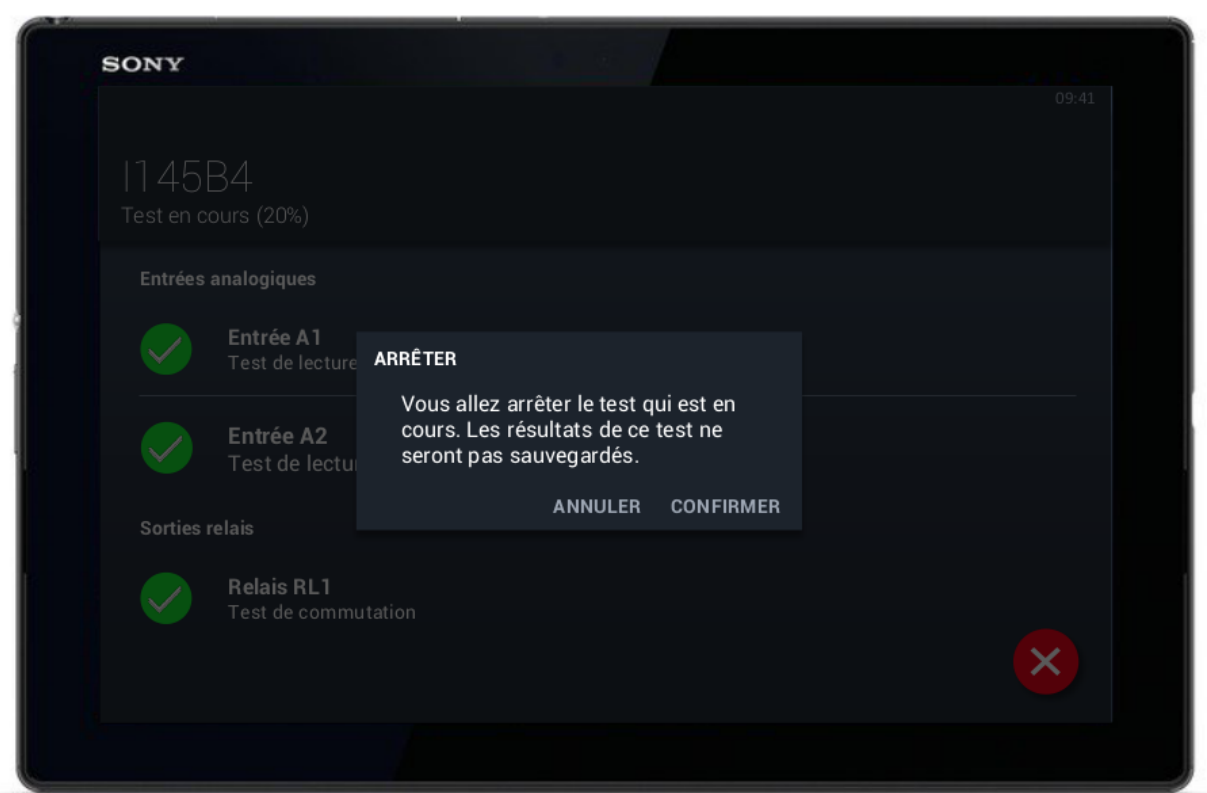


Illustration 47: IHM Dialogue_ConfirmerInterruption


Bouton	Intention / Action
CONFIRMER	valider
ANNULER	annuler
 Retour (bouton tablette)	annuler

Tableau 18: IHM Intention_Dialogue_ConfirmerInterruption

3.1.4.4.7 Dialogue_ErreurEditionTest

Cette boîte de dialogue affiche un message d'erreur lorsqu'un test obligatoire a été désélectionné. Le bouton retour de la tablette ou « FERMER » permettent de fermer la boîte de dialogue.

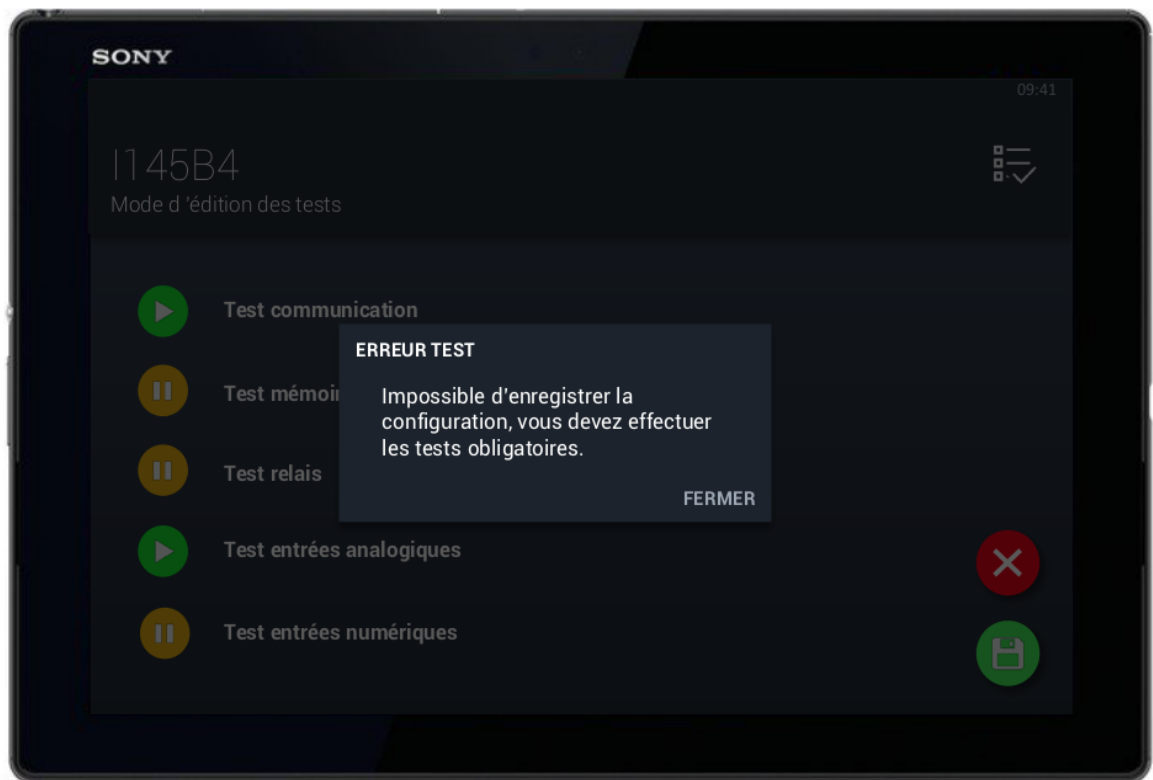


Illustration 48: IHM Dialogue_ErreurEditionTest

Bouton	Intention / Action
FERMER	annuler
Retour (bouton tablette)	annuler

Tableau 19: IHM Intention_Dialogue_ErreurEditionTest

3.1.4.4.8 Dialogue_ErreurInvalide

Cette boîte de dialogue affiche un message d'erreur lorsque l'utilisateur n'a pas sélectionné l'icône permettant de valider ou invalider sur l'écran « Ecran_ResultatTest ». Le bouton retour de la tablette ou « FERMER » permettent de fermer la boîte de dialogue.

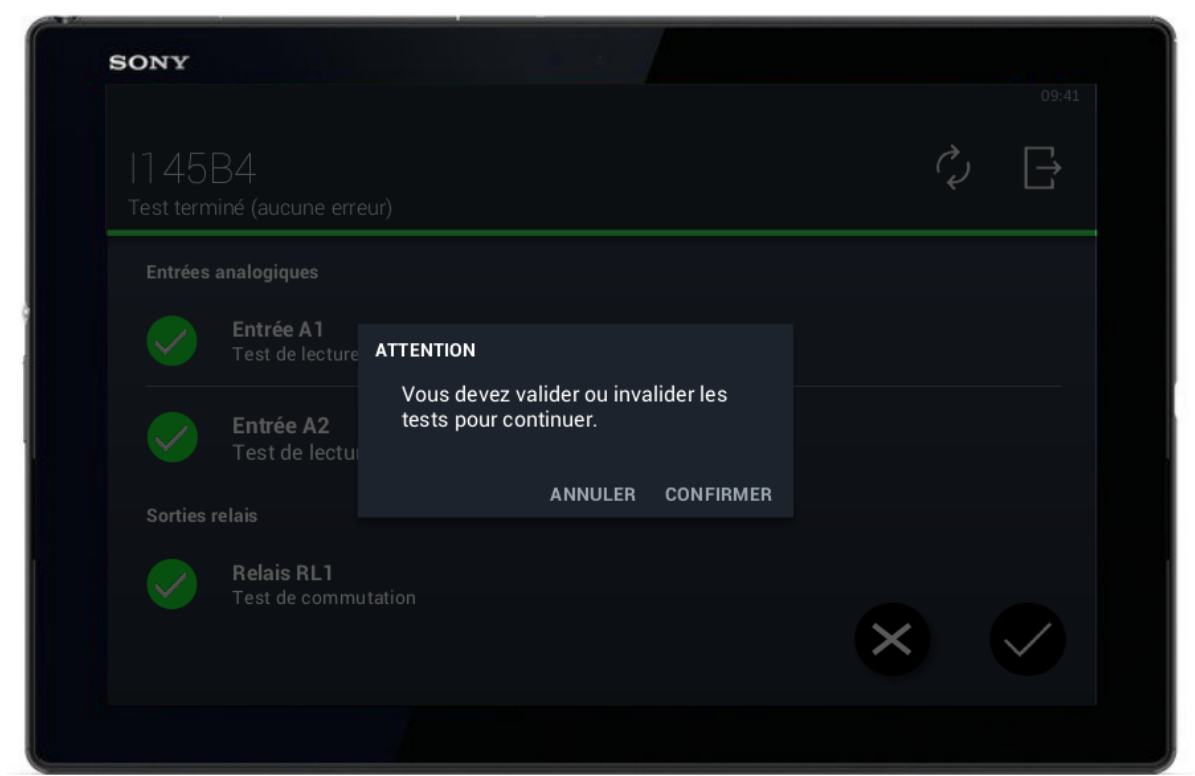


Illustration 49: IHM Dialogue_ErreurInvalide


Bouton	Intention / Action
FERMER	annuler
 Retour (bouton tablette)	annuler

Tableau 20: IHM Intention_Dialogue_ErreurInvalide

3.2 Description des fonctions

3.2.1 Vérifier les identifiants de l'opérateur

Algorithme de "Vérifier les identifiants de l'opérateur"

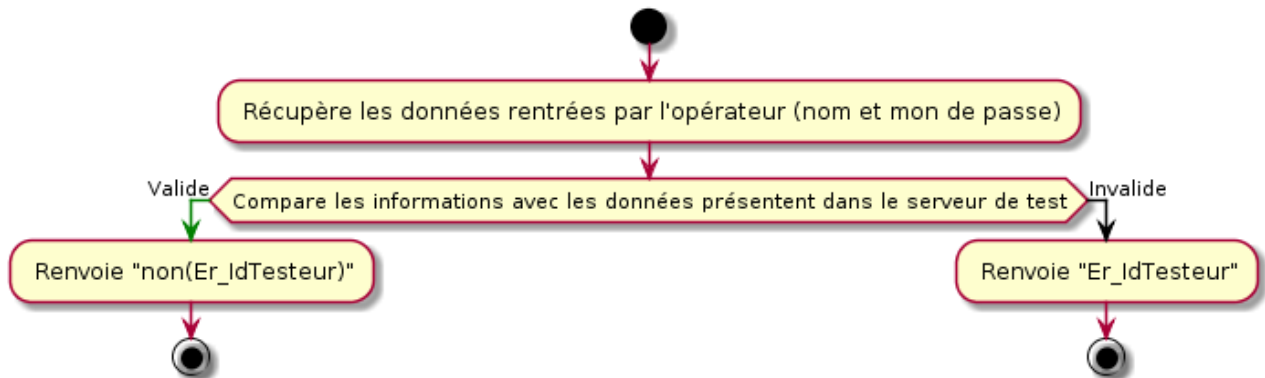


Illustration 50: Algorithme de "Vérifier les identifiants de l'opérateur"

L'Illustration 50 décrit un diagramme d'activité UML l'algorithme de la fonction.

3.2.2 Vérifier le choix des tests à réaliser

Algorithme de "Vérifier le choix des tests à réaliser"

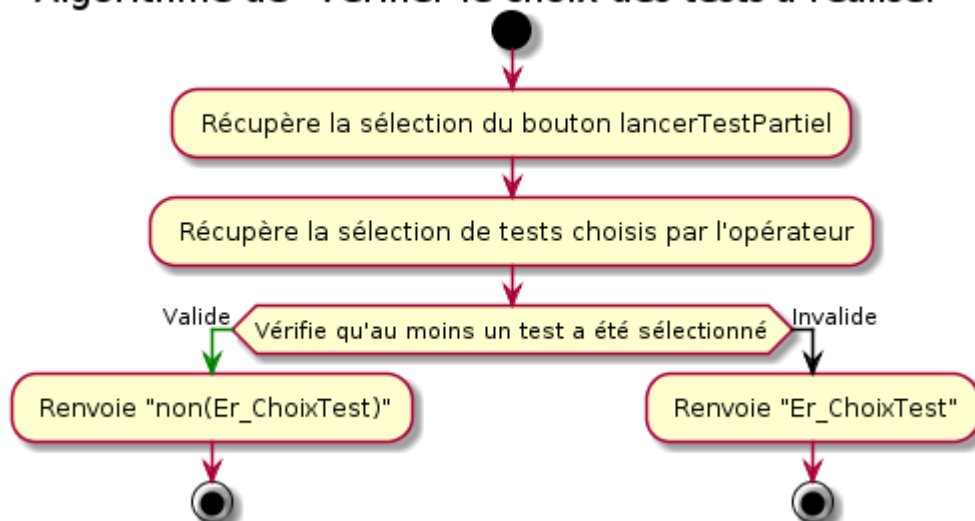


Illustration 51: Algorithme de "Vérifier le choix des tests à réaliser"

3.2.3 Réaliser une sélection de tests

Algorithme de "Réaliser une sélection de tests"

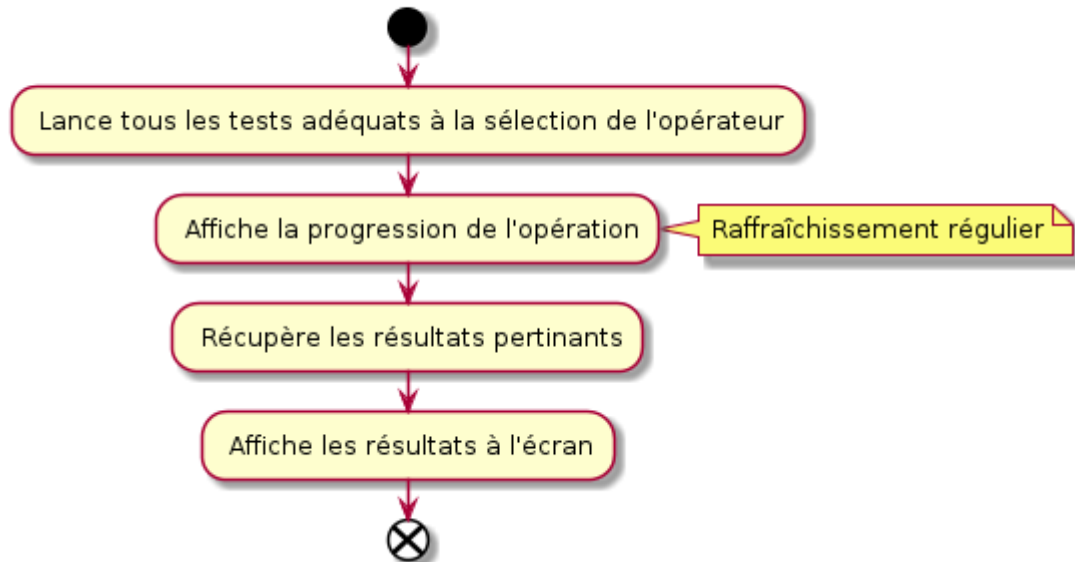


Illustration 52: Algorithme de "Réaliser une sélection de tests"

L'Illustration 52 décrit un diagramme d'activité UML l'algorithme de la fonction.

3.2.4 Consulter l'historique

L'Illustration 53 décrit un diagramme d'activité UML l'algorithme de la fonction.

Algorithme de "Consulter l'historique"

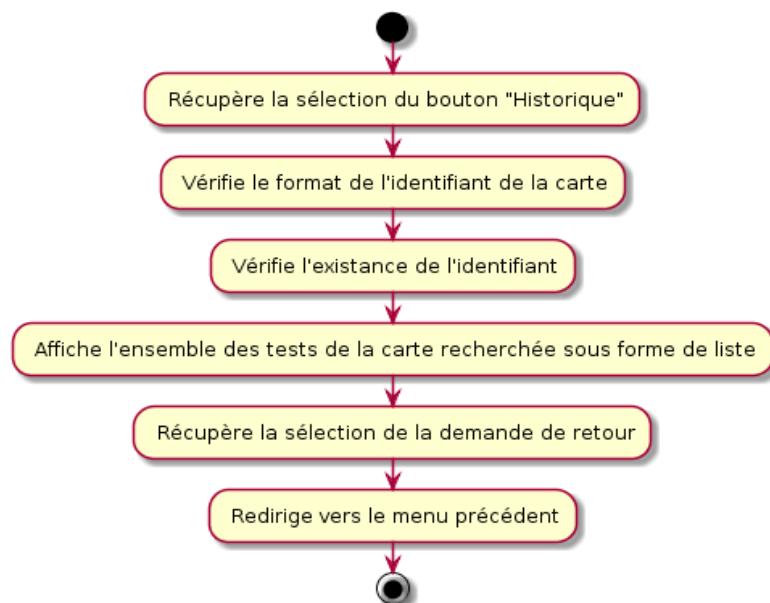


Illustration 53: Algorithme de "Consulter l'historique"

3.2.5 Visualisation des statistiques des tests

L'illustration 54 décrit un diagramme d'activité UML l'algorithme de la fonction.

Algorithme de "Visualisation des statistiques des tests"

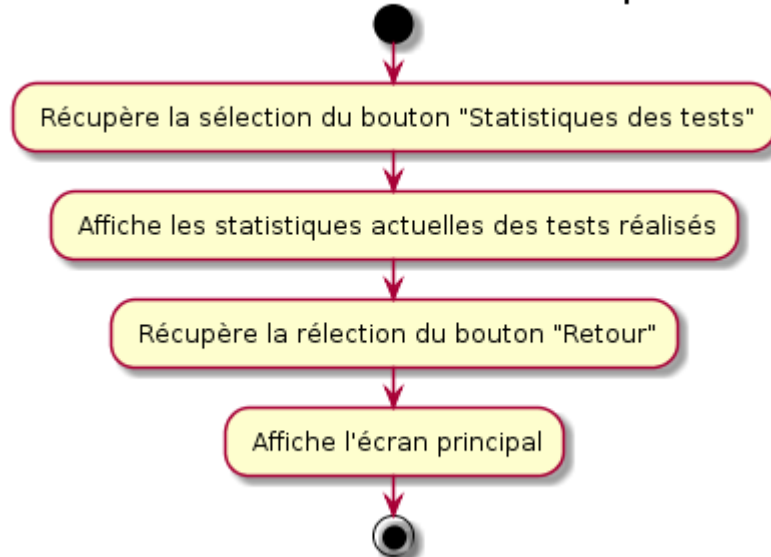


Illustration 54: Algorithme de "Visualisation des statistiques de tests"

3.2.6 Interrompre pour cause d'erreur de communication

L'illustration 55 décrit un diagramme d'activité UML l'algorithme de la fonction.

Algorithme de "Interrompre pour cause d'erreur de communication"

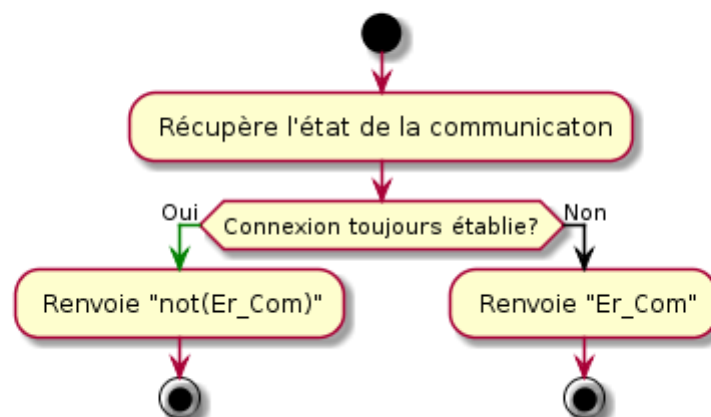


Illustration 55: Algorithme de "Interrompre pour cause d'erreur de communication"

3.2.7 Vérifier la validation du test

L'Illustration 56 décrit un diagramme d'activité UML l'algorithme de la fonction.

Algorithme de "Vérifier la validation du test"

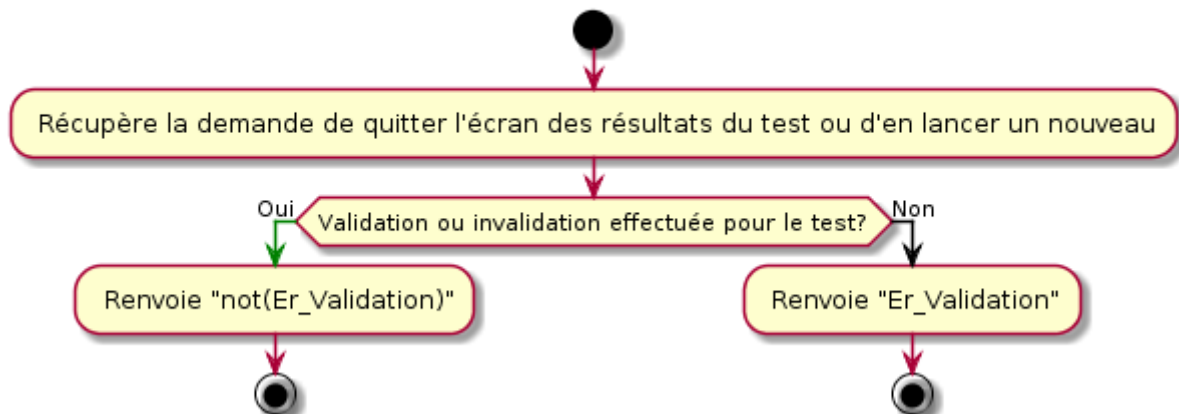


Illustration 56: Algorithme de "Vérifier la validation du test"

3.2.8 Confirmer le choix de quitter l'écran de résultats

L'Illustration 57 décrit un diagramme d'activité UML l'algorithme de la fonction.

Algorithme de "Confirmer le choix de quitter l'écran de résultats du test"

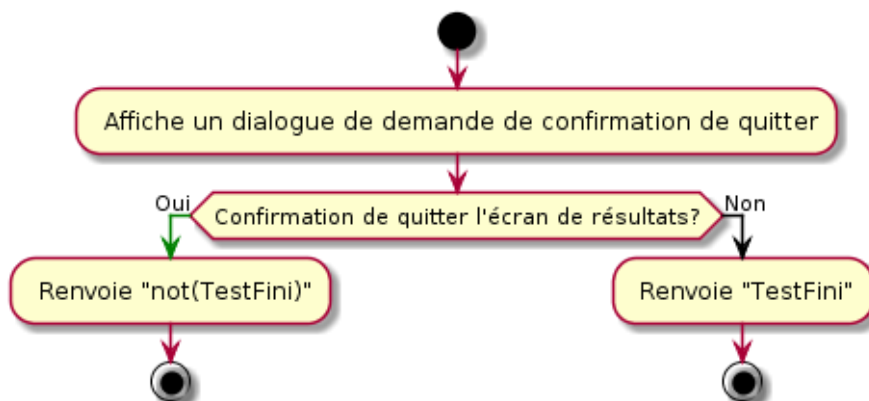


Illustration 57: Algorithme de "Confirmer le choix de quitter l'écran de résultats du test"

3.2.9 Enregistrer les résultats des tests

L'Illustration 58 décrit un diagramme d'activité UML l'algorithme de la fonction.

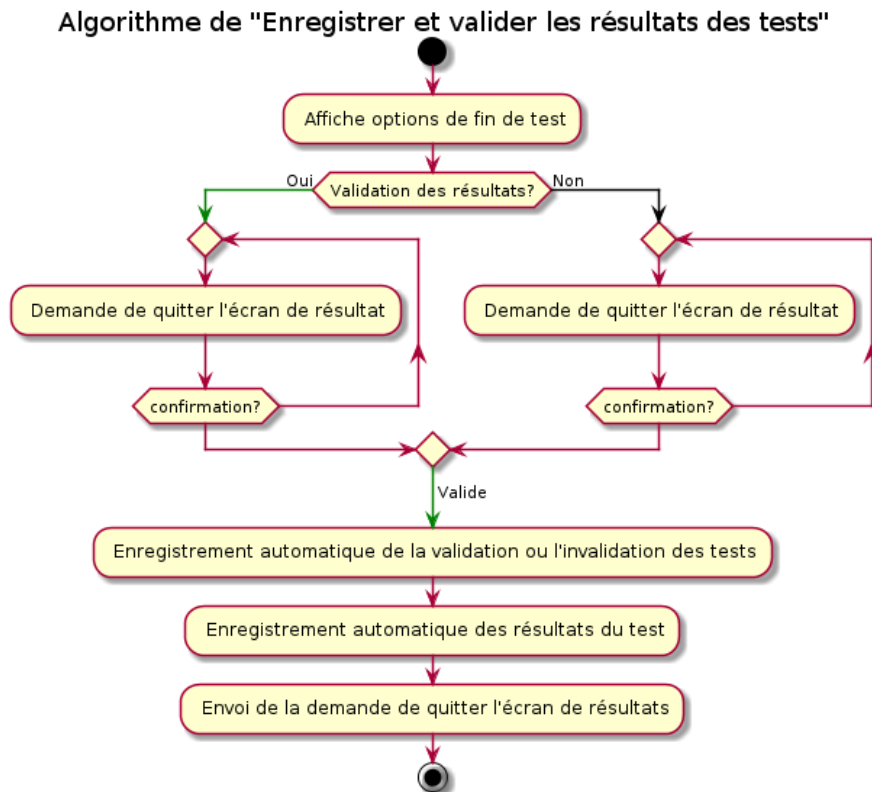


Illustration 58: Algorithme de "Enregistrer et valider les résultats des tests"

3.3 Dictionnaire du domaine

Termes	Définitions
Anciens tests	Les anciens tests affichés relatifs à une carte donnée sont les 10 derniers tests ayant été réalisés avec le produit Thingy sur cette même carte.
Batterie	Composant interne de la tablette fournissant de l'énergie électrique pour le bon fonctionnement de la tablette.
Derniers tests	Les derniers tests sont les 15 derniers tests ayant été réalisé avec le produit Thingy.
En état de fonctionner	Cette expression signifie que : <ul style="list-style-type: none">• La batterie de la tablette est suffisamment chargée.• TE est correctement alimenté.• TA a été correctement chargé sur la tablette.• TC a été correctement chargé sur la RPB. (voir dictionnaire du domaine)
Erreur	Des levées d'erreur peuvent être reportées lors de l'utilisation de Thingy (voir définition) et ainsi éviter tout dégât sur Thingy. Les erreurs sont définies selon le format « Er_MotifErreur » et ont chacune une signification bien spécifique : <ul style="list-style-type: none">• Er_IdTesteur : quand une identification testeur n'a pas pu aboutir.• Er_IdCarte : quand une identification de carte n'a pas pu aboutir.• Er_Com : quand une communication est interrompue.• Er_Interrupt : quand l'application s'interrompt de manière inattendue.• Er_Aliment : quand Thingy n'est plus alimenté.• Er_ChoixTest : Erreur dans le choix des tests (un test obligatoire désélectionné).• Er_Validation : Erreur de validation, l'utilisateur n'a pas validé ou invalidé les résultats.
Est connecté(e)	Cette expression signifie que : <ul style="list-style-type: none">• La tablette est connectée avec succès à TH.• L'ordinateur de développement est connecté avec succès à TH.

	<ul style="list-style-type: none"> • TH est connecté avec succès à l'IPX800V4. • Le serveur de test est connecté avec succès à TH. Le serveur de test peut lire et envoyer les fichiers de transmission .csv .
Est identifié(e)	<p>Cette expression signifie que :</p> <ul style="list-style-type: none"> • L'opérateur est identifié auprès de TA. • La carte à tester est identifiée auprès de TA.
ID_Carte	Représente l'état « carte identifiée ».
ID_Testeur	Représente l'état « carte identifiée ».
IPX800 V4	Appareil autonome fourni par l'entreprise GCE Electronics (voir Client, dictionnaire des abréviations) sur lequel sera réalisé les différents tests produits par Thingy.
Menu principal	Interface principale de l'application TA.
NotReady	L'IHM n'est pas prêt à être utilisé.
Opérateur	Ce terme représente l'utilisateur de Thingy qui décide les tests à réaliser. On parle d' « opérateur » utilisant l'outil logiciel Thingy comme définit dans le cahier des charges [CDCProSe_GCEv2_SANS_EA2].
PC de développement	Ce terme désigne l'ordinateur de développement permettant de simuler directement le comportement du logiciel TC.
Périphériques Thingy	Les périphériques de Thingy sont tous les composants indépendants du système développés par des sociétés autres que ESEO et qui permettent à Thingy de réaliser ses objectifs (tester les cartes IPX800V4). Les périphériques présents sont les suivants : une tablette et un PC de développement.
Raspberry	Carte électronique disposant d'un nano-ordinateur permettant l'intégration du logiciel TC.
Ready	L'IHM est prête à être utilisée
RPB	Ce terme désigne la carte électronique Raspberry sur laquelle opère TC.
SSH	Le SSH (Secure Shell) est un protocole de communication sécurisé.
Statistiques des tests	Les statistiques des tests sont les données statistiques décrivant le taux de réussite des tests selon les appareils.

	Elles permettent de définir la fiabilité des tests.
TA	Ce terme désigne le logiciel interface s'occupant de l'affichage et de la demande de tests (Thingy_App).
TA_EcranTesteur	Temps d'attente entre l'écran de lancement et l'écran d'identification de l'opérateur. Il est estimé à environ 3 secondes.
Tablette	Ce terme désigne la tablette Android utilisée par l'opérateur. Cette tablette supportera le fonctionnement du logiciel TA.
TC	Ce terme désigne le logiciel de contrôle permettant le lancement et la gestion des tests (Thingy_Commande).
TE	Ce terme désigne la carte créée par l'entreprise Prøve qui facilite la communication avec la carte IPX800V4 à tester.
Temps de lancement	Correspond à l'attente du temps d'attente TA_EcranTesteur. (cf dictionnaire du domaine)
Test_Fini	Condition vrai lorsque le test est terminé.
Test obligatoire	Un test obligatoire est un test faisant partie de la liste de tests suivantes : <ul style="list-style-type: none"> • test Ethernet • test du bus de communication • test RJ12 • test NVRAM • test Flash
TH	Ce terme désigne l'ensemble Thingy_Hard (voir dictionnaire du domaine) regroupant les différents dispositifs de la partie hardware du projet Thingy.
Thingy	Ce terme désigne l'ensemble du SaE. (voir dictionnaire des abréviations)
Thingy_App	Logiciel Android permettant l'affichage de l'IHM (voir dictionnaire des abréviations) et la liaison avec TC.
Thingy_Commande	Logiciel embarqué sur la carte RPB permettant l'exécution des tests à réaliser sur le IPX800 V4 et la liaison avec TC.
Thingy_Elektroniks	Ce terme désigne la carte électronique réalisée par l'entreprise Prøve, couplée avec la RPB, qui facilite la mise en œuvre des tests.

Thingy_Hard	Ensemble comportant la carte RPB et TE.
Thingy_Soft	Ensemble logiciel comportant TA et TC.
X8-R	Appareil autonome fourni par l'entreprise GCE Electronics (voir Client, dictionnaire des abréviations) qui permettra de tester la connectivité entre deux appareils GCE Electronics.

Index des illustrations

Illustration 1: Légende du diagramme de déploiement UML.....	14
Illustration 2: Architecture matérielle et logicielle imposée sans TE.....	15
Illustration 3: Architecture matérielle et logicielle imposée avec TE.....	17
Illustration 4: Contexte logique du SaE sans TE.....	20
Illustration 5: Contexte logique du SaE avec TE.....	26
Illustration 6: Contexte physique du SaE sans TE.....	29
Illustration 7: Contexte physique du SaE avec TE.....	30
Illustration 8: Capture d'écran de l'envoi d'un signal de la X8R.....	33
Illustration 9: Capture d'écran de l'envoi d'un signal de la X8R.....	33
Illustration 10: Principaux concepts d'un cas d'utilisation.....	37
Illustration 11: CU stratégique Thingy.....	39
Illustration 12: CU LancerTests.....	43
Illustration 13: IHM Intentions.....	53
Illustration 14: IHM_AppliUsing.....	54
Illustration 15: IHM_Ecrans_Test.....	55
Illustration 16: IHM Ecran_Lancement.....	64
Illustration 17: IHM Dialogue_ErreurConnexion.....	65
Illustration 18: IHM Dialogue_ReconnecterOuQuitter.....	66
Illustration 19: IHM Ecran_IdentificationTesteur.....	67
Illustration 20: IHM Message_IdentificationTesteur.....	68
Illustration 21: IHM Erreur_IdentificationTesteur.....	68
Illustration 22: IHM Ecran_Principal.....	70
Illustration 23: IHM Ecran_HistoriqueCarte (recherche).....	71
Illustration 24: IHM Ecran_HistoriqueCarte (erreur).....	72
Illustration 25: IHM Ecran_HistoriqueCarte (sans détail).....	73
Illustration 26: IHM Ecran_HistoriqueCarte (avec détail).....	74
Illustration 27: IHM Ecran_StatistiqueTest (recherche).....	75
Illustration 28: IHM Ecran_StatistiqueTest (erreur).....	76
Illustration 29: IHM Ecran_StatistiqueTest (avec statistiques).....	76
Illustration 30: IHM Dialogue_TesteurDeconnexion.....	77
Illustration 31: IHM Dialogue_RechercheCarte.....	78
Illustration 32: IHM Dialogue_CreationRecherche.....	79
Illustration 33: IHM Dialogue_ErreurIdCarte.....	80
Illustration 34: IHM Dialogue_ConfirmerQuitter1 (depuis login).....	81
Illustration 35: IHM Dialogue_ConfirmerQuitter2 (depuis home).....	82
Illustration 36: IHM Dialogue_IdNewCarte.....	83
Illustration 37: IHM Dialogue_IdCarte.....	84
Illustration 38: IHM Ecran_AttenteTest.....	85

Illustration 39: IHM Ecran_EditionTest (avant sauvegarde).....	86
Illustration 40: IHM Ecran_EditionTest (pendant sauvegarde).....	87
Illustration 41: IHM Ecran_ProgressionTest.....	88
Illustration 42: IHM Ecran_ResultatTest (avant intention).....	89
Illustration 43: IHM Ecran_ResultatTest (pendant sauvegarde).....	90
Illustration 44: IHM Dialogue_ConfirmerQuitter2 (depuis attente test).....	91
Illustration 45: IHM Dialogue_ConfirmerQuitter2 (depuis résultat test).....	92
Illustration 46: IHM Dialogue_ConfirmerQuitter3 (pendant sauvegarde).....	93
Illustration 47: IHM Dialogue_ConfirmerInterruption.....	94
Illustration 48: IHM Dialogue_ErreurEditionTest.....	95
Illustration 49: IHM Dialogue_ErreurInvalide.....	96
Illustration 50: Algorithme de "Vérifier les identifiants de l'opérateur".....	97
Illustration 51: Algorithme de "Vérifier le choix des tests à réaliser".....	97
Illustration 52: Algorithme de "Réaliser une sélection de tests".....	98
Illustration 53: Algorithme de "Consulter hitorique".....	98
Illustration 54: Algorithme de "Visualisation des statistiques de tests".....	99
Illustration 55: Algorithme de "Interrompre pour cause d'erreur de communication".....	99
Illustration 56: Algorithme de "Vérifier la validation du test".....	100
Illustration 57: Algorithme de "Confirmer le choix de quitter l'écran de résultats du test".....	100
Illustration 58: Algorithme de "Enregistrer et valider les résultats des tests".....	101

Index des tableaux

Tableau 1: IHM Intention_Dialogue_ErreurConnexion.....	65
Tableau 2: IHM Intention_Dialogue_ReconnecterOuQuitter.....	66
Tableau 3: IHM Intention_IdentificationTesteur.....	69
Tableau 4: IHM Intention_Ecran_Principal.....	70
Tableau 5: IHM Intention_Ecran_HistoriqueCarte.....	74
Tableau 6: IHM Intention_Ecran_StatistiqueTest.....	77
Tableau 7: IHM Intention_Dialogue_TesteurDeconnexion.....	78
Tableau 8: IHM Intention_Dialogue_RechercheCarte.....	79
Tableau 9: IHM Intention_Dialogue_ErreurIdCarte.....	80
Tableau 10: IHM Intention_Dialogue_ConfirmerQuitter.....	82
Tableau 11: IHM Intention_Dialogue_IdNewCarte.....	83
Tableau 12: IHM Intention_Dialogue_IdCarte.....	84
Tableau 13: IHM Intention_Ecran_AttenteTest.....	85
Tableau 14: IHM Intention_Ecran_EditionTest.....	87
Tableau 15: IHM Intention_Ecran_ProgressionTest.....	88
Tableau 16: IHM Intention_Ecran_ResultatTest.....	90
Tableau 17: IHM Intention_Dialogue_ConfirmerQuitter2.....	92
Tableau 18: IHM Intention_Dialogue_ConfirmerInterruption.....	94
Tableau 19: IHM Intention_Dialogue_ErreurEditionTest.....	95
Tableau 20: IHM Intention_Dialogue_ErreurInvalide.....	96