



Université du Québec
École de technologie supérieure

LOG410 : ANALYSE ET SPÉCIFICATIONS DU LOGICIEL

Présenté à

DAVID GUILMAINE

Natural Born Coders, J-UI

Vision

David Lauzon
LAUD01028300

Anton Zakharov
ZAKA12038406

18 octobre 2010

Table des matières

1	Introduction	4
1.1	Objectif	4
1.2	Portée	4
1.3	Définitions, acronymes et abréviations	4
1.4	Références	5
1.5	Vue d'ensemble	5
2	Positionnement	5
2.1	Opportunité d'affaires	5
2.2	Énoncé du problème	6
2.3	Positionnement du produit	7
3	Descriptions des intervenants et des utilisateurs	7
3.1	Marché Cible	7
3.2	Résumé des intervenants	7
3.3	Résumé des utilisateurs	7
3.4	Environnement utilisateur	8
3.5	Profils des intervenants	8
3.6	Profils des utilisateurs	8
3.7	Principaux besoins des intervenants et utilisateurs	8
3.8	Alternatives et Compétition	9
4	Vue d'ensemble du produit	9
4.1	Perspective du produit	9
4.2	Principaux avantages	10
4.3	Hypothèses et dépendances	10
4.4	Prévisions de coûts et de prix	11
4.5	Licences et installation	11
5	Caractéristiques du produit	11
5.1	Afficher tous les composants du SBAJ (FEA01)	11
5.2	Connecter les <i>widgets</i> du J-UI aux valeurs des composants de SBAJ. (FEA02) . . .	11
5.3	Sauvegarder une instance de J-UI.(FEA03)	11
5.4	Charger une instance de JUI.(FEA04)	11
5.5	Se connecter à au moins un SBAJ.(FEA05)	11
5.6	Changer la taille d'un <i>widget</i> .(FEA06)	11
5.7	Changer la position d'un <i>widget</i> .(FEA07)	11
5.8	Enlever/Ajouter un <i>widget</i> sur la perspective. (FEA08)	11
5.9	Avoir un ensemble de <i>widgets</i> génériques.(FEA09)	12
5.10	Avoir un ensemble de <i>widgets</i> spécifiques. (FEA10)	12
5.11	Avoir la possibilité d'ouvrir plusieurs instances J-UI à la fois sur le même ordina- teur.(FEA11)	12
5.12	Avoir un système permettant de gérer des alertes.(FEA12)	12

6 Contraintes	12
7 Gammes de qualité	12
8 Attributs des caractéristiques	12
9 Autres exigences du produit	13
9.1 Standards applicables	13
9.2 Exigences du système	13
9.3 Exigences de performance	14
9.4 Exigences environnementales	14
10 Exigences de documentation	14
10.1 Manuel de l'utilisateur	14
10.2 Aide en-ligne	14
10.3 Guides d'installation, de configuration, et fichier à lire	14
10.4 Étiquetage et emballage	15
11 Bibliographie	16
A Annexe A : Attributs des caractéristiques	17

TAB. 1: Historique des Révisions

Date	Version	Description	Auteur
22/sept/2010	v0.1	Ébauche initiale	Anton Zakharov, David Lauzon
10/oct/2010	v0.2	Correction fautes sections 4,6,7	David Lauzon
14/oct/2010	v1.0	Correction revue par les pairs	David Lauzon, Anton Zakharov

1 Introduction

1.1 Objectif

Le but de ce document est de collecter, analyser et définir les besoins haut niveaux et les fonctionnalités de l'application J-UI. Le document va se focaliser dans l'identification des fonctionnalités nécessaires pour les différents participants et pour les utilisateurs directs, en plus de décrire pourquoi ces fonctionnalités doivent exister. Les détails de comment J-UI remplit ces besoins sont spécifiés dans des documents supplémentaires comme les spécifications des exigences logiciel (SRS) et dans les cas d'utilisations (CU).

Ce document a pour objectif d'illustrer d'une manière claire et concise les différents besoins des différents intervenants et les fonctionnalités qui devraient être implémentées pour que le produit soit satisfaisant.

1.2 Portée

Ce document de vision porte sur le développement et l'intégration de J-UI avec le système de contrôle informatique du sous-marin autonome de SONIA. De plus l'outil développé devrait pouvoir être réutilisé avec d'autres projets utilisant le protocole de communication JAUS, comme les logiciels de contrôle pour le sous-marin autonome et le logiciel pour la tondeuse de gazon automatique. Les niveaux de détails pour les fonctionnalités dans le présent document est minimum.

1.3 Définitions, acronymes et abréviations

AUV5

Système décisionnel logiciel présent actuellement sur sous marin autonome SONIA.

AUV6

Mise à niveau de AUV5.

Contrôle

Élément visuel simple (ex : tableau, zone de texte, etc.) qui permet d'afficher / modifier une valeur.

ÉTS

École de Technologie Supérieure.

GUI (Graphic User Interface)

l'interface graphique utilisateur.

JAUS (Joined Architecture for Unmanned Systems)

Architecture logicielle et protocole de communication pour les engins autonomes.

SONIA

Club Étudiant de l'École de Technologie Supérieure entièrement composé de volontaires, des étudiants en ingénierie qui sont dévoués au développement d'un véhicule sous-marin autonome.[1]

SBAJ

Système basé sur l'architecture JAUS. Système dont les composantes communiquent en JAUS et organisés en JAUS (ex : système informatique du sous-marin de SONIA, quand il sera basé sur JAUS).

Composante de SBAJ .

Composante du système, par ex : moteur, torpille, etc.

Source de données

Ensemble de toutes les données de télémétrie disponibles.

Télémétrie

Technologie qui permet la mesure à distance et la journalisation d'informations d'intérêt vers le concepteur du système ou un opérateur.[2]

Valeur

Donnée de télémétrie.

Widget

Élément visuel composé d'un ou plusieurs contrôles et d'une ou plusieurs sources de données et qui peut être déplacé, redimensionné, etc. dans l'écran.

1.4 Références

Club S.O.N.I.A (École de technologie supérieure)

<http://sonia.etsmtl.ca>

Association for unmanned vehicle systems international

<http://www.auvsi.org/AUVSI/AUVSI/Home/>

Document fourni en exemple

Éditeur de Vision de LQD Inc

Cours d'Analyse de besoins et spécifications (ETS)

<https://cours.etsmtl.ca/log410/private/>

1.5 Vue d'ensemble

Cette section ne s'applique pas dans le cadre de ce projet.

2 Positionnement

2.1 Opportunité d'affaires

Cette section ne s'applique pas dans le cadre de ce projet.

2.2 Énoncé du problème

Le problème de	L'interface (GUI) de télémétrie actuelle est désuète et partiellement fonctionnelle :
affecte	<ul style="list-style-type: none"> – La communication entre JAUS et la télémétrie actuelle est mal supportée. – La perspective (dimensions, positions des widgets) n'est pas sauvegardée entre les sessions de télémétrie (ouverture et fermeture du logiciel). – Le changement d'une valeur dans l'interface de télémétrie ne garantit pas le changement dans le paramètre du sous-marin. – Il serait compliqué de mettre à jour la télémétrie pour supporter le AUV6.
dont l'impact est	<p>Les membres de l'équipe de SONIA</p> <ul style="list-style-type: none"> – Difficulté d'implémenter les mises à jour dans l'interface de télémétrie. – Impossibilité de communiquer avec les versions futures de AUV, donc rendant inutile l'application présente. – Perte de temps pour les utilisateurs du sous-marin lors des phases de tests et en compétition. – Possibilité de mauvaise lecture de valeur – Erreur critique reliée a une valeur modifiée dans l'interface mais pas envoyée au sous-marin. – Perte de temps pour le service de maintenance logicielle pour implémenter les rustines (<i>patch</i>).
Une bonne solution serait	Concevoir une nouvelle interface de télémétrie depuis le début (<i>from scratch</i>) indépendante de AUV et basée sur JAUS (une technologie plus stable). De plus l'interface serait configurable par chacun des utilisateurs.

Le problème de	Plusieurs clubs étudiants implémentent une télémétrie utilisant JAUS mais leurs interfaces GUI utilisent les mêmes <i>widgets</i> mais ne sont pas ré-utilisable d'un club a l'autre.
affecte	Les étudiants des clubs utilisant une télémétrie basée sur JAUS (Dronalab et Capra par exemple).
dont l'impact est	<ul style="list-style-type: none"> – Problème de maintenance. – Duplication des efforts, programmation en double.
Une bonne solution serait	Implémenter une interface basée sur JAUS permettant la ré-utilisation de <i>widgets</i> communs. Aussi, il faudrait qu'il soit possible pour les clubs de créer leurs propres <i>widgets</i> .

2.3 Positionnement du produit

Pour	Pour les utilisateurs de télémétrie basée sur JAUS, de différents clubs étudiants.
Qui	Veulent un produit stable, facile a utiliser et maintenir.
Le J-UI	est une interface de télémétrie
Qui	Est fiable, hautement configurable, facilement maintenable et amicale a l'utilisateur (<i>user-friendly</i>).
Contrairement à Notre produit	Une interface difficile a utiliser, lourde a maintenir, non-fiable. Sauve du temps précieux aux étudiants en leur offrant un produit fiable, hautement configurable, facilement maintenable et amicale a l'utilisateur (<i>user-friendly</i>).

3 Descriptions des intervenants et des utilisateurs

3.1 Marché Cible

Cette section ne s'applique pas dans le cadre de ce projet.

3.2 Résumé des intervenants

Nom	Description	Responsabilités
1) Capitaine du club. (Kevin Larose)	Administrateur responsable des développements des projets.	Détermine si la version de télémétrie peut être utilisée pour la compétition présente ou non. Demande des changements dans la télémétrie.
2) Utilisateur J-UI (Simon Bolduc)	Utilise le logiciel de télémétrie pour les besoins du projet de chaque club.	S'assure du bon fonctionnement de chaque projet de chaque club grâce au logiciel de télémétrie.
3) Programmeurs J-UI (Francois Campeau)	Personne qui crée et qui maintient un logiciel.	Gens responsables de la télémétrie actuelle et future et des widgets compatibles avec JAUS.

Note : les noms sont sujets a changements, voila pourquoi les titres des postes sont mentionnés.

3.3 Résumé des utilisateurs

Nom	Description	Responsabilités	Intervenant
Utilisateur de J-UI	Tous les membres de SONIA, Capra, et Dronalab qui utilisent la télémétrie. Dans le cas qu'il faudrait spécifier pour le projet, le nom du projet apparaîtrait après le nom de l'utilisateur.	<ul style="list-style-type: none">– Choisir les bonnes métriques affichées.– S'assurer que les valeurs des métriques représentées affichent les valeurs réelles par rapport aux automates des projets.– S'assurer que les valeurs changées dans la télémétrie sont bien changées dans le robot.	Intervenant 2 dans la liste des intervenants.

3.4 Environnement utilisateur

L'environnement de l'utilisateur est un ordinateur (soit laptop ou ordinateur de bureau) avec clavier et souris. Pour ce qui est de SONIA, le programme de télémétrie est un programme important/critique. Au point que si une défaillance se produit il faudrait interrompre les tests et beaucoup de temps précieux serait perdu. Le programme doit fonctionner dans les environnements Windows, Linux et Mac.

3.5 Profils des intervenants

Cette section ne s'applique pas dans le cadre de ce projet.

3.6 Profils des utilisateurs

Cette section ne s'applique pas dans le cadre de ce projet.

3.7 Principaux besoins des intervenants et utilisateurs

TAB. 2: Besoins

Besoin	Priorité	Préoccupations	Solution actuelle	Solution proposée
Gérer la disposition, sélection des widgets de la perspective.	Critique	L'utilisateur veut pouvoir sélectionner quels <i>widgets</i> sont présents sur la perspective et comment ils sont disposés (taille, position).	L'interface n'offre pas la possibilité ni de choisir, ni d'ajouter des <i>widgets</i> additionnels dans l'écran. De plus, il est impossible de changer la position.	L'utilisateur vas pouvoir sélectionner quels <i>widgets</i> seront présents sur la perspective et comment ils seront disposés (taille, position).
Pouvoir sauvegarder/charger la perspective des <i>widgets</i> .	Critique	Chaque utilisateur de la télémétrie a des données qui sont utiles pour lui. Il veut pouvoir configurer les <i>widgets</i> en conséquence.	L'interface est la même pour tout le monde. Tous les changements faits à l'interface sont perdus quand on ferme le logiciel.	L'utilisateur vas pouvoir sauvegarder la perspective de son interface dans J-UI. La prochaine fois qu'il utilise l'interface il va avoir la possibilité de charger l'interface sauvegardée précédemment.

TAB. 2: (suite)

Besoin	Priorité	Préoccupations	Solution actuelle	Solution proposée
Pouvoir ouvrir plusieurs interfaces de télémétrie simultanément.	Important	L'utilisateur veut avoir plusieurs interfaces de télémétrie ouvertes en même temps, car c'est utile pour visualiser différentes données de l'avancement du sous marin.	On ne peut avoir qu'une seule interface ouverte.	L'utilisateur vas pouvoir ouvrir autant d'interfaces de J-UI qu'il veut.
Gérer des alertes si une valeur critique est atteinte par un <i>widget</i> .	Important	Un utilisateur veut savoir immédiatement si certaines valeurs sont atteintes pour des composantes critiques. (ex: batteries vides).	Présentement une simple icône, montre l'état des batteries.	Une alerte affichée dans J-UI qu'il est possible de combiner avec n'importe quel <i>widget</i> . L'alerte apparait si une valeur critique est atteinte.
Ajouter facilement d'autres <i>widgets</i> .	Important	Un programmeur veut pouvoir ajouter des <i>widgets</i> « <i>in house</i> » (conçu sur mesure) facilement. Ceci est pratique si une nouvelle composante avec des besoins spécifiques est ajoutée.	Il faut complètement « <i>hard coder</i> » un nouveau <i>widget</i> . Il faut re-tester tout la télémétrie pour pouvoir l'intégrer.	Le J-UI est un système flexible dans lesquels on peut facilement ajouter des nouveaux <i>widgets</i> .

3.8 Alternatives et Compétition

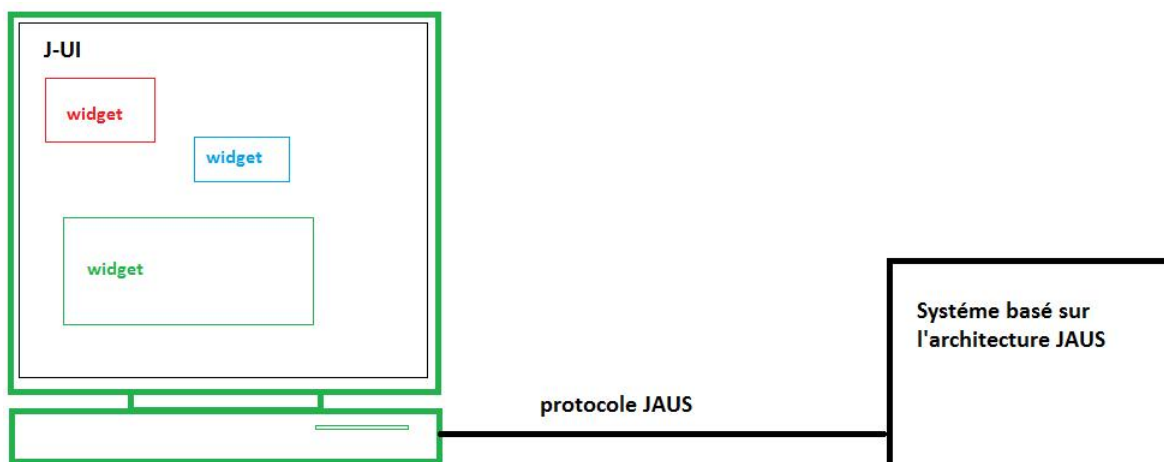
Cette section ne s'applique pas dans le cadre de ce projet.

4 Vue d'ensemble du produit

4.1 Perspective du produit

Le J-UI vas pouvoir être utilisé avec n'importe quel projet communiquant par le protocole JAUS et supportant une architecture JAUS. Le J-UI est une interface utilisateur (GUI) permettant d'afficher des widgets. Les widgets affichent et permettent de modifier les données des composantes qui communiquent avec le J-UI. Les composantes doivent supporter une architecture JAUS. Étant donné que le commanditaire principal de J-UI est le club sous marin SONIA certain *widgets* spécifiques seront créées enfin de pouvoir supporter les besoins particuliers de celui-ci.

Le J-UI pourrait être configuré par des utilisateurs experts et utilisé par des utilisateurs normaux. Il vas marcher sur une ordinateur portable ou un *desktop* possédant une communication avec un système basé sur l'architecture JAUS.



4.2 Principaux avantages

TAB. 3: Avantages J-UI

Bénéfices pour le client	Caractéristiques correspondantes
Gérer la disposition, sélection des widgets de la perspective.	FEA01, FEA02, FEA06, FEA07, FEA08
Pouvoir sauvegarder/charger la perspective (dimensions, position) des <i>widgets</i> .	FEA03, FEA04
Pouvoir ouvrir plusieurs interfaces de télémétrie ouvertes simultanément.	FEA05, FEA11
Ajouter des alertes si une valeur critique est atteinte par un <i>widget</i> .	FEA12
Ajouter facilement d'autres <i>widgets</i> .	FEA09, FEA10

4.3 Hypothèses et dépendances

1. Nous présumons que le club SONIA possède un ordinateur portable qui supporte le J-UI.
2. Nous présumons que le club SONIA possède un SBAJ qui puisse communiquer avec l'ordinateur, nommé dans le point 1, utilisant le protocole JAUS.
3. Nous présumons que à défaut de se conformer au point 2. que le club SONIA possède une interface pouvant émuler le comportement d'un SBAJ et d'une communication avec un SBAJ.
4. Nous présumons que SONIA possède déjà une librairie permettant la communication utilisant JAUS.

4.4 Prévisions de coûts et de prix

Cette section ne s'applique pas dans le cadre de ce projet.

4.5 Licences et installation

Le J-UI est la propriété de l'ÉTS. Le club SONIA et d'autre clubs étudiants peuvent utiliser librement le logiciel, si l'ÉTS le permet.

Le J-UI serait installé manuellement par un membre de club SONIA sur un ordinateur portable ou un *Desktop*. Le J-UI serait disponible sur les plateformes Windows, Linux et Mac.

5 Caractéristiques du produit

5.1 Afficher tous les composantes du SBAJ (FEA01)

Permettrait de voir toutes les composantes du SBAJ dont on peut modifier / afficher les valeurs. Cela permettra a chacun des membres des clubs d'agencer leur(s) perspective(s) a leur guise.

5.2 Connecter les *widgets* du J-UI aux valeurs des composantes de SBAJ. (FEA02)

Les *widgets* du J-UI vont afficher une ou plusieurs valeurs des composantes de SBAJ et vont permettre de les modifier. Or pour ça il va falloir qu'ils puissent communiquer d'une façon ou d'une autre avec les composantes de SBAJ.

5.3 Sauvegarder une instance de J-UI.(FEA03)

Permet de sauvegarder l'interface J-UI, cad. la taille, position des *widgets*.

5.4 Charger une instance de JUI.(FEA04)

Permet de charger une instance de J-UI sauvegardée précédemment.

5.5 Se connecter à au moins un SBAJ.(FEA05)

Le J-UI doit avoir la possibilité de se connecter à un ou plusieurs SBAJ.

5.6 Changer la taille d'un *widget*.(FEA06)

Le J-UI doit offrir la possibilité de changer la taille d'un *widget* ou cette action à du sens.

5.7 Changer la position d'un *widget*.(FEA07)

Le J-UI doit offrir la possibilité de changer la position d'un *widget* ou cette action à du sens.

5.8 Enlever/Ajouter un *widget* sur la perspective. (FEA08)

Le J-UI doit offrir la possibilité pour un utilisateur de choisir quels *widgets* seront présents sur une perspective donnée.

5.9 Avoir un ensemble de *widgets* génériques.(FEA09)

Le J-UI aura plusieurs *widgets* pouvant être ré-utilisés d'un club a un autre (ex : tableau, boutons, zone de texte, etc.).

5.10 Avoir un ensemble de *widgets* spécifiques. (FEA10)

Il doit être possible a chacun des clubs d'implémenter leurs widgets spécialisés pour leur projet (ex : visualiseur de profondeur du sous-marin, direction, etc).

5.11 Avoir la possibilité d'ouvrir plusieurs instances J-UI à la fois sur le même ordinateur.(FEA11)

Le J-UI doit permettre l'ouverture de plusieurs instances sur la même machine. Toutes les instances peuvent se connecter sur la même/différente SBAJ.

5.12 Avoir un système permettant de gérer des alertes.(FEA12)

L'utilisateur peut assigner une alerte (message) a une valeur critique. Lorsque cette valeur critique (ex : 10% de batterie restante) a été atteinte, l'utilisateur reçoit un message qui attire son attention.

6 Contraintes

1. Le J-UI pour démontrer le maximum des ce capacités doit pouvoir être connecté à un SBAJ ou à une interface émulant le comportement de ce dernier.
2. Le J-UI doit être développé avec le langage de programmation Java.
3. Communiquer à travers un protocole JAUS avec un SBAJ.
4. Rendre les *widgets* modulaires, indépendants de J-UI.

7 Gammes de qualité

Le J-UI doit être facile d'utilisation et l'interface doit être efficace pour l'utilisateur. Étant donné que les utilisateurs sont des experts inclure des raccourcis. Le J-UI doit être tolérant aux fautes (ex : si une composante du sous marin *crash* le J'UI ne doit pas *crasher*). Le J-UI doit être modulaire et maintenable. Ceci permetterat d'ajouter facilement d'autres *widgets*.

8 Attributs des caractéristiques

TAB. 4: Attributs des caractéristiques

Caractéristiques	État	Bénéfice	Effort	Risque	Stabilité	Priorité
Afficher tous les composants du SBAJ (FEA01)	Proposé	Élevé	Faible	Faible	Moyen	Critique

TAB. 4: Attributs des caractéristiques

Caractéristiques	État	Bénéfice	Effort	Risque	Stabilité	Priorité
Connecter les <i>widgets</i> du J-UI aux valeurs des composantes de SBAJ.(FEA02)	Proposé	Élevé	Élevé	Élevé	Élevé	Critique
Sauvegarder une instance de J-UI.(FEA03)	Proposé	Élevé	Moyen	Faible	Faible	Important
Charger une instance de JUI.(FEA04)	Proposé	Élevé	Moyen	Faible	Faible	Important
Se connecter à au moins un SBAJ.(FEA05)	Proposé	Élevé	Moyen	Moyen	Moyen	Critique
Changer la taille d'un <i>widget</i> .(FEA06)	Proposé	Moyen	Faible	Faible	Faible	Important
Changer la position d'un <i>widget</i> .(FEA07)	Proposé	Élevé	Faible	Faible	Faible	Important
Enlever/Ajouter un <i>widget</i> sur la perspective. (FEA08)	Proposé	Moyen	Faible	Faible	Faible	Critique
Avoir un ensemble de <i>widgets</i> génériques.(FEA09)	Proposé	Élevé	Élevé	Faible	Faible	Critique
Avoir un ensemble de <i>widgets</i> spécifiques. (FEA10)	Proposé	Élevé	Élevé	Élevé	Élevé	Critique
Avoir la possibilité d'ouvrir plusieurs instances J-UI à la fois sur le même ordinateur.(FEA11)	Proposé	Moyen	Faible	Faible	Faible	Important
Avoir un système permettant de gérer des alertes.(FEA12)	Proposé	Moyen	Faible	Faible	Moyen	Utile

9 Autres exigences du produit

9.1 Standards applicables

ESA1. La programmation logicielle doit suivre le standard de programmation de Java de Sun.

ESA2. Les standards de programmations doivent être conformes avec les standards de programmation du club SONIA.

9.2 Exigences du système

ES1. Performances minimales de l'ordinateur exécutant la télémétrie :

- Processeur : Pentium IV 1 GHz.
- Mémoire : 2 Gb de mémoire RAM.
- Un clavier est une souris.

ES2. Le système opérationnel doit être au minimum :

- Windows XP service pack 2.
- Windows Vista service pack 1.
- Windows 7 service pack 1.

- MacOS X.
 - Ubuntu 8.04.
- ES3. Une possibilité d'établir une connexion avec un SBAJ. Ceci peut être :
- Une carte réseau qui peu être connecté au SBAJ.
 - Un émulateur de connexion au SBAJ.

9.3 Exigences de performance

EP1. Le temps de réponse de changement de la télémétrie doit être au plus une seconde a part si spécifié autrement par l'utilisateur du J-UI. En d'autre mots si une valeur est changée dans le SBAJ, elle doit prendre au plus une seconde pour être affichée correctement dans le J-UI. Si une valeur est changée par le J-UI le changement dans le SBA doit être effectué en moins de 50 ms.

EP2. Le temps d'établissement de connexion du J-UI au SBAJ ne doit pas dépasser 5 secondes.

9.4 Exigences environnementales

- EE1. Le temps moyen de recouvrement du J-UI doit être de 10 secondes.
- EE2. L'interface doit avoir un contraste élevé pour avoir une bonne visibilité sous le soleil.

10 Exigences de documentation

ED1. Les documents techniques seront produits en français. Le manuel de l'utilisateur doit être écrit au minimum en français et en anglais. Les documents techniques seront disponibles seulement en ligne. Le manuel de l'utilisateur serait disponible en français.

10.1 Manuel de l'utilisateur

EMU1. Le manuel de l'utilisateur complet aurait pour but de permettre à l'utilisateur de configurer, utiliser, sauvegarder et charger l'interface J-UI. De plus il serait décrit comment établir une connexion avec un SBAJ. Un index et un glossaire seront nécessaires.

EMU2. Un manuel rapide serait crée, qui contiendra les raccourcis et les informations/commandes les plus importantes.

10.2 Aide en-ligne

- EAL1. L'aide en ligne va contenir la version informatique des deux manuels.
- EAL2. L'aide en ligne vas posséder une fonction de recherche, d'index et d'une table des matières avec des hyperliens permettant de rechercher rapidement l'information.
- EAL3. L'aide en ligne va contenir un petit tutoriel de comment configurer et utiliser le J-UI.
- EAL4. L'aide en ligne va contenir des captures d'écran avec des informations pertinentes.
- EAL5. Les Widgets supplémentaires qui seront ajoutés plus tard vont avoir leurs descriptions incluses dans l'aide en ligne.

10.3 Guides d'installation, de configuration, et fichier à lire

EGCF1. Un petit «Read Me» serait disponible pour montrer comment installer le logiciel. Mais comme le logiciel est assez simple l'installation pourrait être faite par un Wizard.

10.4 Étiquetage et emballage

Cette section ne s'applique pas dans le cadre de ce projet.

11 Bibliographie

- [1] *Page d'accueil*. 5 Oct. 2010. Club Étudiant SONIA. Visité en Octobre 2010.
<http://sonia.etsmtl.ca/fr>
- [2] *Télémetrie (informatique)*. 10 fév. 2010. Wikipédia. Visité en Octobre 2010.
[http://fr.wikipedia.org/wiki/T%C3%A9l%C3%A9m%C3%A9trie_\(informatique\)](http://fr.wikipedia.org/wiki/T%C3%A9l%C3%A9m%C3%A9trie_(informatique))

A Annexe A : Attributs des caractéristiques

TAB. 5 – État

Proposé	La caractéristique est proposée, mais n'a pas encore été approuvée par les parties prenantes.
Approuvé	La caractéristique est approuvée par les parties prenantes.
Incorporé	La caractéristique est incluse dans le produit.

TAB. 6 – Bénéfice

Faible	La caractéristique apporte peu de valeur ajoutée au produit et n'est pas nécessaire à son bon fonctionnement.
Moyen	La caractéristique apporte une valeur ajoutée additionnelle au produit, mais n'est pas critique à son bon fonctionnement.
Élevé	La caractéristique apporte une valeur ajoutée importante au produit et est essentielle à son bon fonctionnement ou à la réalisation de ses tâches.

TAB. 7 – Effort

Faible	La réalisation de la caractéristique nécessite un effort de moins de 20 heures-personnes.
Moyen	La réalisation de la caractéristique nécessite un effort entre 20 et 40 heures-personnes.
Élevé	La réalisation de la caractéristique nécessite un effort de plus de 40 heures-personnes.

TAB. 8 – Risque

Faible	La technologie utilisée et la méthode d'implémentation sont connues et bien maîtrisées.
Moyen	La technologie utilisée est récente ou la méthode d'implémentation nécessite une attention particulière.
Élevé	La technologie utilisée est nouvelle et peu éprouvée ou la méthode d'implémentation est complexe et demande une analyse plus complète.

TAB. 9 – Stabilité

Faible	Les exigences concernant la caractéristique ont peu de chance de changer et le bon fonctionnement de la caractéristique n'a pas d'impact sur le fonctionnement général du système.
Moyen	Les exigences concernant la caractéristique sont susceptibles de changer ou le bon fonctionnement de la caractéristique a un impact sur le fonctionnement général du système sans toutefois compromettre son exécution.
Élevé	Les exigences concernant la caractéristique ont de fortes chances de changer ou le bon fonctionnement de la caractéristique a un impact critique sur le fonctionnement général du système et peut compromettre son exécution.

TAB. 10 – Priorité

Faible	La caractéristique apporte des fonctionnalités accessoires au système. Son inclusion dans le produit a peu d'impact sur la satisfaction du client et sur l'utilisation du système.
Moyen	La caractéristique apporte des fonctionnalités supplémentaires au système. Son inclusion dans le produit peut influencer la satisfaction du client, mais son absence n'empêche pas l'utilisation du système.
Élevé	La caractéristique est primordiale au fonctionnement du système. Il est nécessaire de l'inclure en priorité dans le produit pour assurer la totale satisfaction du client et son absence pourrait empêcher l'utilisation du système.