

**Département de génie logiciel et des T.I.**

Rapport de Laboratoire

|  |  |
| --- | --- |
| **Numéro du laboratoire** | 04 |
| **Nom du laboratoire** | Projet de session |
| **Étudiant(s)** | Jean-Simon Bonin  Xavier Duval  Martin Gingras |
| **Code(s) permanent(s)** | BONJ02098701  DUVX20048607  GINM10108708 |
| **Cours** | LOG 430 |
| **Session** | Hiver 2012 |
| **Groupe** | 01 |
| **Chargé(e) de laboratoire** | Samir Djeffal |
| **Date** | 16 avril 2012 |

Table des matières

[Introduction 5](#_Toc322013890)

[Développement 6](#_Toc322013891)

[Discussion 7](#_Toc322013892)

[Conclusion 8](#_Toc322013893)

**Liste des tableaux**

**No table of figures entries found.**

**Liste des figures**

**No table of figures entries found.**

# Introduction

Ce laboratoire a pour objectif de nous permettre d’effectuer la conception d’une architecture d’un système en se basant sur un document de vision, un SRS et des cas d’utilisation tout en suivant l’ensemble des méthodologies vues dans le cours d’architecture logicielle. La réalisation des exigences de ce laboratoire mettra en pratique ces diverses méthodologies et servira de récapitulatif complet pour ce cours.

La suite de ce rapport contient trois sections. D’abord, la section principale du document, le développement, contient l’ensemble de notre conception architecturale. La deuxième section est une discussion comparant les quatre laboratoires sur lesquels nous avons eu à travailler cette session et faisant état de notre point de vue sur ce quatrième laboratoire. Une conclusion vient clore le document en faisant un retour sur les objectifs du laboratoire.

# Développement

**Rappel de la nature et la mission commerciale du système**

Le système est un client de télémétrie qui communique par l’entremise du protocole JAUS et qui gère un ensemble de widgets, perspectives et alertes. Le système permet de créer des widgets qui présenteront les données acquis par les différents senseurs externes au client qui seront interrogés à l’aide de la librairie libJAUS. De plus, un ensemble de widgets dans une configuration donnée peut être enregistré sous forme de perspective. Il y a donc des perspectives qui peuvent être créé pour chaque club étudiant ou utilisateur du client. Le système permet aussi la création d’alertes sur certaines valeurs des senseurs.

La mission du système est de profiter non seulement pour le club étudiant SONIA mais aussi pour tout autre club étudiant pour autant qu’il respecte le protocole de JAUS. De plus, le système doit être multiplateforme et être développé en JAVA.

L’architecture de notre système est déterminée par certains attributs de qualité plus dominants soit la maintenabilité, l’interopérabilité, l’utilisabilité et la portabilité.

**Présentation de l’architecture**

La présentation de l’architecture contient au moins une vue de chacun des trois types principaux (modules, composant et connecteurs, affectation).

La présentation de votre architecture devrait identifier les approches utilisées et expliquer comment les attributs de qualité jugés importants sont supportés par votre solution. Aussi, vous devez analyser au moins six scénarios en détail, et pour chacun produire un tableau d'analyse détaillé tel que vu au cours. Vous devez naturellement produire plus loin quatre tableaux, explicitant respectivement les risques, points de compromis, points de sensibilité et non-risques.

Vue de type modules - décomposition:

****

Documentation des modules

|  |  |
| --- | --- |
| **Module** | LibJAUS |
| **Responsabilités** | Ce module est responsable de la communication entre les capteurs et les autres modules du système. |
| **Cas d’utilisations supporté** |  |
| **Interfaces** | Interface qui fournit des services qui permet d’obtenir les valeurs des différents capteurs et périphériques externe. |
| **Scénarios supporté** |  |
| **Contraintes** |  |
| **Données produit** |  |
| **Données requis** |  |

|  |  |
| --- | --- |
| **Module** | LiveUV.Controller.WidgetManager |
| **Responsabilités** | Ce module est responsable |
| **Cas d’utilisations supporté** |  |
| **Interfaces** |  |
| **Scénarios supporté** |  |
| **Contraintes** |  |
| **Données produit** |  |
| **Données requis** |  |

**Identification des approches architecturales**

Pour ce qui est des approches architecturales, nous avons utilisé certaines tactiques afin de bien soutenir nos attributs de qualités. En effet, pour l’utilisabilité, on permet à l’utilisateur de créer ses propres perspectives et de les enregistrés.

**Analyse des approches architecturales**

**Arbre d’utilité**

Il doit y avoir au moins quinze (15) scénarios dans votre arbre

-Utilité

-Maintenabilité

-Ajout de nouveau widget

-La nouvelle fonctionnalité est codée en moins de 4 heures.

-Importer un nouveau widget

-L’importation est effectué en moins de 2 heures.

-Mettre à jour l’interface libJAUS

-La mise à jour nécessite moins qu’un mois par 1 personne.

-Interopérabilité

-Support multiplateforme

-Utilisabilité

-Possible de sauvegarder les préférences d’un utilisateur

-Existe des perspectives par default pour chaque club étudiant

-Portabilité

-Compatible pour plusieurs types de véhicules

Il est très important que les attributs de qualité soient bien élaborés sous forme de scénarios.

**Scénarios de qualité**

# Discussion

Text

# Conclusion

Ce laboratoire avait pour objectif de