

Note de cadrage

I) Introduction

Dans le cadre de notre formation en IUT informatique, nous travaillons avec l'entreprise TMM groupe sur un projet de client SNMP sous Android.

L'entreprise TMM est le leader en France dans le domaine des logiciels de santé à usage des soignants et des patients. Ils sont implantés partout en France mais aussi à l'étranger, comme en Belgique ou en Allemagne. TMM crée des solutions d'accompagnement et de divertissement dans le milieu hospitalier disponibles sur différents supports (tablettes numériques, box TV...) le tout sous le système d'exploitation Android. Dans le cadre de l'amélioration de leurs produits et de la capacité d'intervenir rapidement en cas de problème sur leur parc grandissant, l'entreprise souhaite le développement d'un service Android, installé sur tous les dispositifs de la marque, utilisant le protocole SNMP pour envoyer à un serveur Centreon des « TRAP » (messages d'erreur du protocole SNMP) indiquant que le dispositif rencontre un problème technique.

Le service devra récolter un maximum d'informations utiles concernant le problème technique et les transmettre au serveur Centreon afin de trouver la cause du problème (problème lié à l'utilisateur, au logiciel ou au matériel) pour que TMM puisse améliorer son offre logicielle.

II) La portée et les limites de l'étude

- Les utilisateurs

L'application étant autonome elle ne possédera pas d'utilisateur physique quand elle fonctionnera sur le parc. En revanche, lorsque nous testerons l'application sur les différents terminaux de TMM, nous utiliserons une interface graphique de débogage. Les utilisateurs avec qui l'application va communiquer seront donc:

- Le serveur Centreon: il est autonome, son but est de traiter les TRAP que les terminaux clients enverront.
- L'API Android : son rôle est de permettre l'accès aux informations à propos des composants physiques mais aussi logiciels de la tablette afin de déceler des problèmes techniques.
- L'API de l'application TMM : c'est l'interface de l'application qui nous permettra de vérifier l'état de ses modules (téléphone...).
- Le débogueur : c'est l'acteur utilisant l'interface, qui ne sera pas accessible dans la version finale, qui permettra de déboguer l'application sur les différents terminaux de l'entreprise (lancer manuellement les test, envoyer manuellement des TRAP).

- Objectifs utilisateurs :

- Analyser le terminal pour trouver des problèmes
 - Détecter des problèmes de type hardware
 - Détecter des problèmes de type software
- Envoyer des données au serveur
 - Préparer des TRAP
 - Emettre des TRAP
- Déboguer l'application

Les charges utilisateurs

- Gérer le débogage : Lancer manuellement les différentes analyses traitées par le service et retourner le succès ou l'échec du test. Permettre d'envoyer manuellement le résultat du test dans une TRAP.

Les charges du système

- Gérer l'analyse du terminal: l'application devra être autonome pour savoir si elle est en mode ROOT et ne pas analyser trop souvent le matériel ce qui causerait un ralentissement de la tablette.
- Gérer l'analyse du Hardware : l'application devra utiliser l'API Android pour vérifier si le matériel est opérationnel et ne présente pas d'utilisation anormale.
- Gérer l'analyse du software : l'application devra déceler les dysfonctionnements possibles au niveau logiciel sans en créer de nouveaux.
- Gérer la préparation de TRAP: L'application devra mettre en forme les TRAP pour contenir l'identification du problème et des informations le concernant.
- Gérer la préparation de TRAP: L'application devra mettre en forme les TRAP pour contenir l'identification du problème et des informations le concernant.
- Gérer l'envoi de TRAP: L'application devra envoyer les TRAP du protocole SNMP en utilisant le réseau du terminal.

- Le contenu

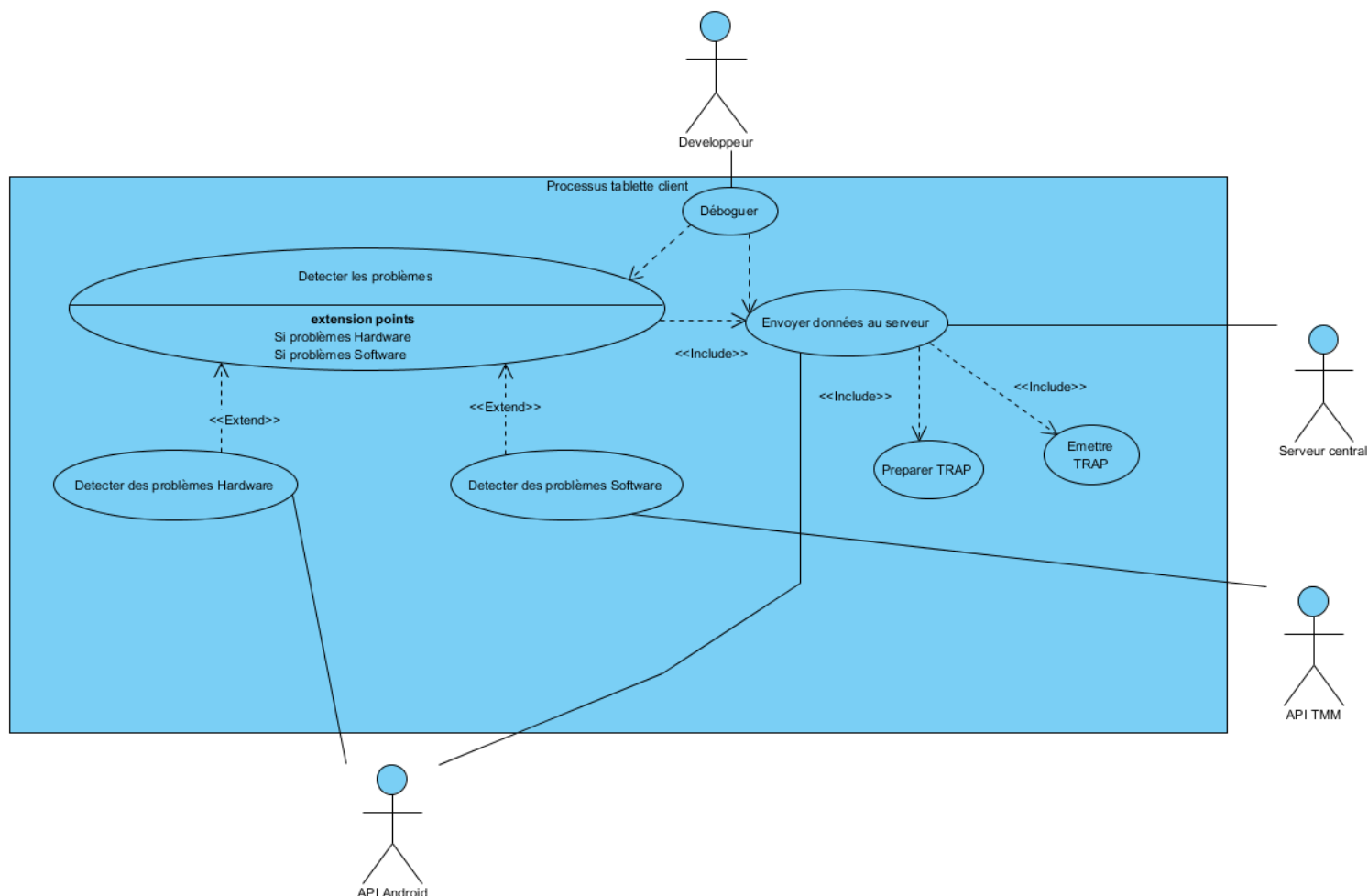
Pour le bon déroulement de l'application principale de la tablette cliente, nous allons devoir créer un processus invisible, léger portatif sur différents matériels et architectures.

Son rôle sera d'analyser la tablette et de détecter des problèmes qui peuvent être tout aussi bien hardware (surchauffe du processeur, utilisation anormale de la RAM, arrêt inopiné de la tablette) que software (fermeture de l'application principale, dysfonctionnement de certaines applications comme le téléphone ou la télécommande). Quand un de ces problèmes est rencontré, notre application enverra une TRAP SNMP au serveur central qui sera géré par l'entreprise TMM et qui interviendra aussitôt sur le problème.

Notre application sera développée en JAVA et utilisera le protocole SNMP.

Dans le cadre du bon développement de notre application, nous allons devoir créer une surcouche qui va posséder une interface graphique et nous permettra de déboguer notre programme. Grâce à cela nous pourrons voir étape par étape si notre application principale (qui ne possédera en version finale aucune interface) réagira correctement aux attentes que nous aurons fixées et si elle pourra accéder aux API nécessaires pour les test (dans le cas où nous ne sommes pas ROOT par exemple).

- Le diagramme de cas d'utilisation



- **Détail des cas d'utilisations**

Détecter les problèmes :

- Préconditions : la tablette est lancée, l'application TMM est lancée, notre application est lancée.
- Scénario : On recherche si on est ROOT puis on lance toutes les détections de problèmes possibles, en fonction du mode ROOT ou non, de façon récurrente. Si on nous retourne un problème on lance l'envoi des données au serveur avec les informations collectées sur l'origine du problème.
- Postconditions : le service est fermé, la tablette s'éteint.

Détecter des problèmes Hardware:

- Préconditions : Le cas Détecter les problèmes nous lance une demande de vérification hardware.
- Scénario : On recherche un problème de type Hardware grâce à l'API Android (état de la RAM, du CPU, de la ROM et du réseau). Si on a découvert un problème matériel on collecte des informations à son propos et on les prépare pour les remonter.
- Postconditions : les informations collectées sont remontées, la vérification est terminées.

Détecter des problèmes Software:

- Préconditions : Le cas Détecter les problèmes nous lance une demande de vérification software.
- Scénario : On recherche un problème de type Software grâce à l'API TMM (état des modules de TMM, application TMM toujours allumée, état de l'application TMM), si on a découvert un problème software on collecte des informations à son propos et on les prépare pour les remonter.
- Postconditions : les informations collectées sont remontées, la vérification est terminées.

Envoyer données au serveur:

- Paramètres : Un ensemble d'informations relatives au problème rencontré.
- Préconditions : Un problème a été découvert et une demande d'envoi de données à été initiée
- Scénario : On lance la préparation d'une TRAP avec nos données, puis on envoi la TRAP retournée.
- Postconditions : On sait si la TRAP a été envoyée ou non.

Préparer TRAP :

- Paramètres : Un ensemble d'informations relatives au problème rencontré.
- Préconditions : Un problème a été rencontré, des informations relatives au problème ont été transmises
- Scénario : On met les informations en forme pour qu'elles soient contenues dans le format d'une TRAP. On retourne cette TRAP.
- Postconditions : La TRAP est prête et retournée.

Émettre TRAP :

- Paramètre : Une TRAP SNMP.
- Préconditions : Une TRAP a été créée, on nous demande d'envoyer la TRAP au serveur central.
- Scénario : On émet la TRAP, si elle est bien envoyée on retourne un succès si-non on retourne un échec.
- Postconditions : Le résultat a été retournée.

Déboguer:

- Préconditions : Notre application est lancée en mode débogage
- Scénario : lancement manuel des vérifications, affichage des résultats des vérifications (effectuée avec son résultat ou non effectuée avec la cause), préparation et envoi d'une TRAP SNMP au serveur central.
- Postconditions : Débogage terminé, application arrêtée ou passée en mode service

- Les risques

- L'accès au ROOT : certains terminaux possédés par TMM ne sont pas accessible en mode ROOT, ce qui peut empêcher d'accéder à certains aspects de l'API d'Android ou à des processus.
- Fermeture de notre application : en cas de fermeture de notre application pour une raison quelconque, la tablette cliente ne pourra plus émettre de TRAP SNMP et donc déceler des problèmes sur le terminal.
- Application trop lourde : si c'est le cas elle ralentira la tablette et empêchera le bon fonctionnement de l'application TMM.
- Installation universelle : le parc de terminaux géré par TMM étant très larges et composé de différente version d'Android, marques et d'architecture de processeurs. Il faudra donc créer une application universelle.
- Serveur coupé : en cas d'une coupure du serveur principal pour une maintenance ou pour une autre raison, des TRAP SNMP pourront être perdues.
- Coupure de réseau : en cas d'une coupure du réseau, on doit pouvoir à la reprise de la connexion, détecter les problèmes arrivés entre temps.