Cahier des charges

Système domotique : SmartHouse

Présentation

SmartHouse est une solution de domotique ayant été réalisée lors du cours de génie logiciel assuré par M. Vernois. La particularité de la solution se trouve dans son abstraction de la communication avec les matériels pilotés et son ouverture pour le développement des drivers par les fabricants des matériels.

Objectifs

Créer une application web (Java EE) et une application mobile (Android) permettant de visualiser et de piloter les différents équipements d'une maison.

On ne cherche pas à créer la maison, les pièces et installer de nouveaux équipements. Cet étape est à faire en amont (via une application bureau dédiée à ce processus de configuration).

Par mesure de simplification pour le projet, nous nous limitons volontairement à un serveur correspondant à une maison (nous ne gérons donc pas le cas d'un système domotique pour hôtel).

Points fonctionnels

L'application web et mobile ont les mêmes fonctionnalités avec des vues différentes (afin d'être en adéquation avec le moniteur).

- visualiser / sélectionner les différentes maisons;
- visualiser / sélectionner les différentes pièces d'une maison;
- visualiser / sélectionner / piloter les différents équipements d'une pièce (permet de voir l'état des équipements);
- gérer les scénarios (tâches + planifications);

Répartition des tâches

Florent :

- Préparation de l'environnement Java EE :
    - Architecture EAR [WAR + EJB] sur JBoss Server 7.1
    - PrimeFaces en implémentation JSF
- Intégration framework existant en EJB avec passage sur JBoss Hibernate en JTA
- Développement couche service commune (EJB + WebService)
- Refonte du chargement et de la gestion des drivers dans l'environnement Java EE
- Amélioration de la meta-description des actions dans les drivers

Antoine :

- Réalisation d'un POC selon les compétences d'un MOA-CG14.
- Réflexion sur les IHM Web
- Intégration maquette HTML en template JSF.

Alexandre :

- Réflexion sur les IHM Android
- Développement de l'IHM Android
- Création des interfaces générales de parcours de la hiérarchie d'une maison.

Julien :

- Passage en JPA dans le framework : création d'une nouvelle impl DAO.
- Intégration des web-services sur Android par android-ws-client (kSoap testé en amont)
- Implémentation du schéma de navigation android
- Création des interfaces de visualisation et de contrôle d'un équipement

IHM ANDROID - JEE

Cinématique d'utilisation :

1ère étape : **Fenêtre de login**

Authentification (nom d'utilisateur, mot de passe).

2e étape : **Fenêtre d'accueil**

Bandeau en haut : Logo à gauche faisant office de lien vers cette même page (accueil).

Tuiles pour les maisons.

3e étape : **Maison sélectionnée**

En haut à gauche, le logo de l'application qui fait lien vers la page d'accueil.
En haut à droite, un bouton pour switcher vers les scénarios.
Système d'onglet pour chaque zone.
Dans le corps de l'application/page web, on a la liste des pièces (avec une image de la pièce par ex). Quand on clique sur cette image, cela déroule les périphériques dynamiquement, avec comme information l'état du périphérique et un bouton pour piloter le périphérique. Si l'utilisateur clique sur ce bouton, il ouvre une nouvelle fenêtre relative au pilotage du périphérique.

3e étape bis : **Pilotage de périphérique**

Cette nouvelle fenêtre, plus petite que la fenêtre principale, montre le descriptif d'un périphérique, et offre des possibilités pour le piloter;

4e étape : **Switch vers les scénarios**

En haut à gauche, le logo de l'application qui fait lien vers la page d'accueil.
En haut à droite, un bouton pour switcher vers les zones.

Système d'onglets pour chaque scénario.

Dans le corps de l'application/page web, on a la liste des tâches (avec une image pour chaque tâche par exemple). Quand on clique sur une image, cela déroule les actions, puis les triggers. Pour chaque action ou trigger, on peut modifier/supprimer.

Technologies

- Serveur d'application JBoss AS 7.1
- Serveur de base de données MySQL 5

- JBoss Hibernate 4.2 en JTA Provider (mode CMT avec 1 cache level)
- PrimeFaces en JSF Provider
- JBoss Hibernate Validator en Bean Validator (JSR 303)
- Weld en CDI Provider
- Apache CXF en JAX-WS Provider

Critiques et améliorations

Pour des raisons de temps, des éléments sont éloignés des objectifs du projet. Cette liste non exhaustive a pour vocation de montrer notre esprit critique envers notre solution.

Techniquement

- Sécurisation des Web Services par HTTPS
- Authentification systématique ou par token (session) par le biais de JAAS
- Configuration d'un second niveau de cache pour Hibernate avec Infinispan

Fonctionnellement

- Ajout d'un système d'écoute des équipements ayant la faculté d'invoquer le système domotique (exemple alarme).

Gestion de projet

- Outils de qualité