

Documentation Technique

J.O.I.E : Just Order Instantly Easily

Aymeric DERAZEY, Florent LE COZ, Antoine MINEAU, Thomas TEIXEIRA-FREDERIQUE,
Jacky WONG
15/11/2011

Ce document a pour but de présenter le projet JOIE : un système de prise de commande pour tablette tactile, principalement à destination des restaurants, pouvant également être utilisé par d'autres types d'établissement (hôtel, bar, ...). Tout d'abord nous détaillerons le contexte du projet, puis une présentation, et enfin exposerons les aspects techniques : le protocole et les technologies utilisés. La révision du projet correspondant à ce document est la révision 193 sur le SVN.

Description du document

Titre	Documentation technique
Date	15 / 11 / 2011
Auteurs	Antoine MINEAU, Aymeric DERAZEY, Florent LE COZ, Thomas TEIXEIRA, Jacky WONG
E-Mail	joie_2012@labeip.epitech.eu
Sujet	Documentation destinée aux développeurs
Version	2.7

Révisions

Date	Version	Section	Commentaire
04/01/11	1		Première version
04/07/11	2		Ajout du lien vers la doc doxygen, tableau de version, correction orthographe
21/04/11	2.1	1, 2	Ajout de la section 1, 2
28/04/11	2.2	2.1, 3.1	Ajout de la description du schéma, section 3.1, modification du schéma
30/04/11	2.3		Diverses corrections orthographiques, modification du résumé, ajout partie 3.2 et 3.3
09/06/11	2.4	2.2, 3.1	Rajout d'un tableau de comparaison des produits existants sur le marché, description plus détaillée de la base de données, suppression de la sous-partie interfaces, amélioration design, correction

			orthographique
15/07/11	2.5	1.2	Suppression de l'explication de PSGXS &Prosody, ajout de l'explication d'Openfire
14/09/2011	2.6		Màj template
15/11/2011	2.6.1		Correction d'orthographe

Sommaire

1.	Contexte du projet.....	1
1.1.	Rappel.....	1
1.1.1.	Objectif de l'EIP	1
1.1.2.	Organisation du projet	1
1.2.	Environnement	2
1.2.1.	Réalisation	2
1.2.2.	Technologies.....	2
1.2.3.	Matériels	3
1.2.4.	Contraintes	3
2.	Présentation du projet	4
2.1.	Principe de base	4
2.2.	Existant sur le marche	6
3.	Présentation technique	6
3.1.	Description de la base de données	7
3.2.	Librairies Réseau	7
3.3.	Index.....	8

1. Contexte du projet

1.1. Rappel

1.1.1. Objectif de l'EIP

Une des étapes fondamentales dans le cursus de l'Epitech pour former les étudiants à devenir de vrais professionnels est la réalisation d'un projet par groupe de 5 étudiants minimum sur une durée de 18 mois appelé l'EIP (Epitech innovative project). L'EIP permet d'acquérir une forte expérience dans la gestion d'un projet dans son intégralité, et d'offrir un avant goût de la vie professionnelle.

Dans notre cas, l'objectif de notre EIP est le développement d'une solution complète facilitant la gestion des commandes dans les restaurants, hôtels, etc. Diverses vues seront disponibles : côté client (affichage du menu, commande, appel d'un serveur), côté serveur (facturation, service, prise de commande traditionnelle), côté direction (administration des comptes, diverses statistiques, etc).

1.1.2. Organisation du projet

Notre groupe est divisé de tel sorte :

- partie réseau : le-coz_f, mineau_a et teixei_t
- partie graphique : deraze_a et wong_j
- documentation, web : mineau_a

1.2. Environnement

1.2.1. Réalisation

Afin de mener à bien ce projet, nous utiliserons le logiciel de version Mercurial couplé avec CodingTeam, ce dernier permet d'avoir une vue du dépôt en ligne ainsi que divers outils pratiques pour voir l'avancement, l'évolution des bugs... De plus, plusieurs d'entre nous avons eu à l'utiliser de façon professionnelle durant notre stage ainsi que pour le développement de divers projets au sein même de l'école.

En fonction du système d'exploitation, nous utiliserons pour la programmation Gedit/Emacs/Visual Studio.

1.2.2. Technologies

Nous utiliserons uniquement du python (langage multi-plateforme), ainsi que les librairies :

SleekXMPP est une librairie pour Python principalement centré sur la facilité d'utilisation de toutes les extensions XMPP (XEP) en tant que plugins. Nous utilisons le XEP-0060 (Publications-Souscriptions). Le protocole XMPP permet de créer des nœuds sur un serveur pubsub et de publier des informations sur ces nœuds ; un événement est transmis à toutes les entités ayant souscrit au nœud. Ainsi lorsque qu'un garçon de café prend une commande à une table, celle-ci est envoyé à tous les garçons de café s'occupant de cette table. Si l'administrateur décide de mettre à jour la carte des plats, le serveur envoie la nouvelle carte à toutes les interfaces graphiques. Il n'y donc aucune manipulation à faire pour mettre à jour la carte.

Openfire est un serveur XMPP écrit en Java disposant d'une interface web entièrement administrable, de nombreuses extensions sont disponibles dont Pubsub.

1.2.3. Matériels

Afin de mener à bien ce projet, nous le testerons sur divers écrans tactiles ainsi qu'un serveur afin d'héberger notre programme.

On peut également imaginé d'autres outils comme des table tactile ou même des bornes. Pour le moment, cependant, seul la tablette tactile est retenue, cette dernière se rapprochant le plus d'un menu traditionnel.

1.2.4. Contraintes

GTK n'est pas l'outil optimal afin d'obtenir une interface utilisateur ergonomique et agréable. De ce fait, pour plus de flexibilité, nous créons nous même nos widgets, à destination de l'interface des clients.

Les tablettes actuellement en vente sur le marché ne sont pas idéales pour la restauration. En effet, il y a le problème d'autonomie et de recharge de l'appareil, ainsi que le problème de taille modifiant de manière significative la lisibilité de l'application.

2. Présentation du projet

2.1. Principe de base

L'architecture de Joie est la suivante : Client / Serveur. Le serveur sera une machine distante qui s'occupera du réseau entre les interfaces ainsi que la base donnée.

La partie Client sera composée de 3 interfaces différentes en python/GTK, une pour chaque type d'individu :

- le consommateur, utilisé pour commander des plats.
- la personne s'occupant du service en salle, utilisé pour gérer le menu, les serveurs et les tables.
- l'administration où l'on retrouvera les statistiques, ajouter/modifier/supprimer la carte.

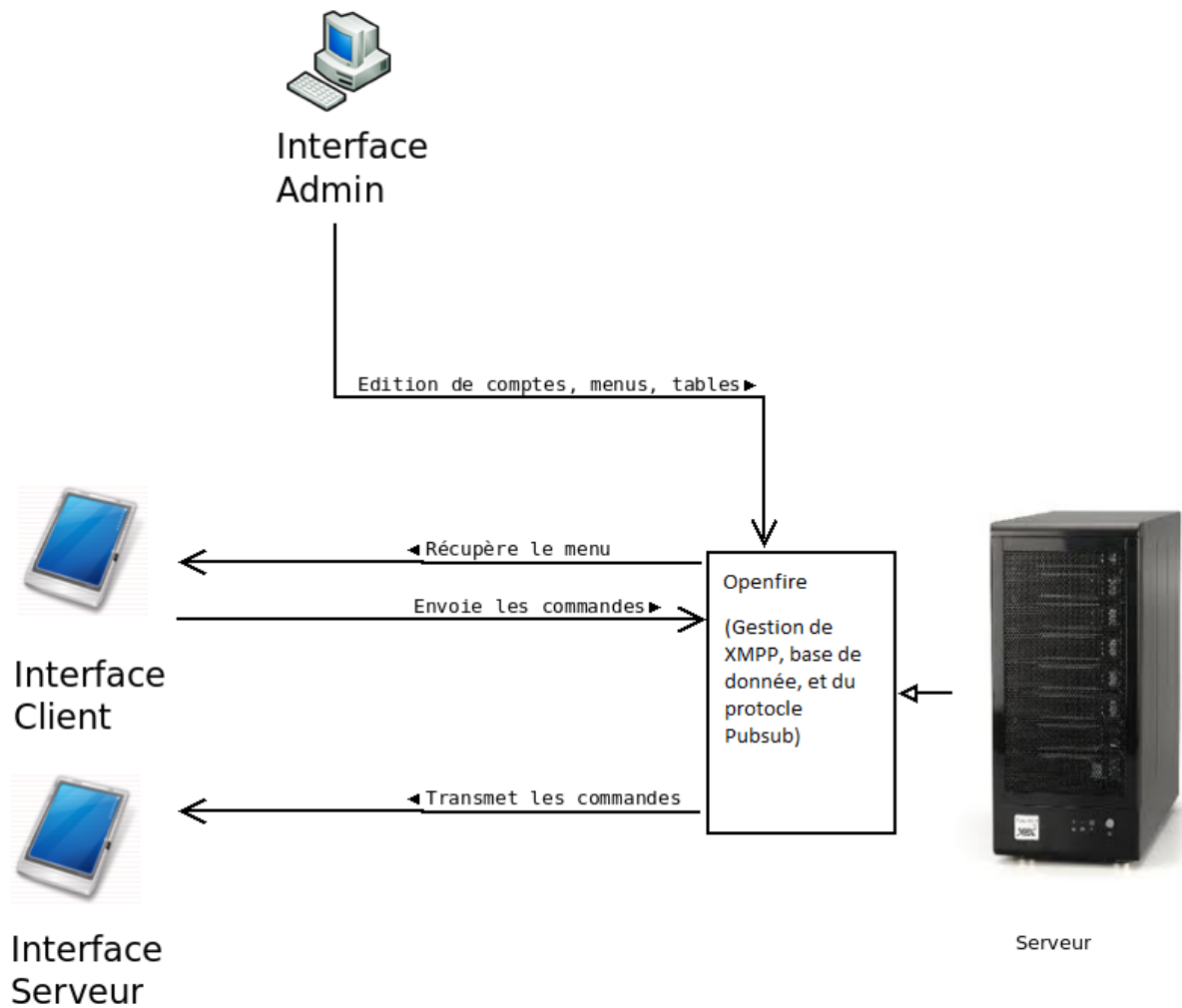


Figure 1. Schéma de fonctionnement de J.O.I.E

La mise en place et utilisation de JOIE est simple:

- L'administrateur crée la carte (plats, menus, boissons...), nombre de tables ainsi que des comptes pour les serveurs de son établissement. Il est nécessaire de lier les serveurs à des tables afin de recevoir les événements de la table (commande, demande de facture...)
- Le serveur donne une interface client
- Une fois la commande passée, cette dernière est directement envoyée au serveur, qui peut l'imprimer ensuite afin de la transmettre en cuisine.

- Lorsque les clients décident de payer, le serveur est prévenu et peut imprimé le ticket listant toutes les commandes qui ont été effectués.

2.2. Existant sur le marche

Des projets semblables au notre existent déjà sur le marché :

Fonctionnalité et Spécificité	Borne KFC	TChooZ	e-tab	JOIE
Mobilité	Non	Non	Oui	Oui
Possibilité de régler à l'aide du dispositif	Oui	Non	Oui	Non
Facilite la commande	Oui	Oui	Oui	Oui
Ajout de fonctionnalité en plus de la commande	Prise et édition de photos	Vote, Messages à d'autres tables	Diviser la note, personnaliser la commande, envoie de la note par mail	Statistiques pour le gérant
Marché visé	Restauration rapide	Bar	Restaurant	Restaurant, bar, hotel

3. Présentation technique

La description des classes et méthodes est disponible sur:
<http://eip.epitech.eu/2012/joie/doxygen/annotated.html>

3.1. Description de la base de données

La base de données est gérée en interne par Openfire. Elle se divise en 3 tables :

- Les plats avec leurs ingrédients, prix, etc.
- Les comptes des garçons de café
- Les tables
- Les commandes passées et en cour

Exemple du champ « content » d'un plat

```
<dish xmlns="xmpp:joie:ns:menu" name="Bo-bun" price="10" short_name="D3"
instock="true">
  <comment>The best dish ever.</comment>
  <categories>
    <category>Vietnam</category>
    <category>Middle dish</category>
  </categories>
  <ingredients>
    <ingredient>Nem</ingredient>
    <ingredient>Vermicelle</ingredient>
  </ingredients>
  <img>
    ...
    [image jpeg ou png zippé puis encodé en base64]
    ...
  </img>
</dish>
```

3.2. Bibliothèques Réseau

Pour les différents clients, développés en Python, nous utilisons la bibliothèque SleekXMPP, à laquelle nous ajoutons un plugin (xep_0060.py) permettant la gestion de PubSub.

Quelques exemples de stanza du protocole pubsub:

```
<iq type='set'
  from='table1@joie.localhost'
```

```
to='pubsub.joie.localhost'  
id='publish1'  
<pubsub xmlns='http://jabber.org/protocol/pubsub'>  
  <publish node='table1'>  
    <item id='bnd81g37d61f49fgn581'>  
      [...]  
    </item>  
  </publish>  
</pubsub>  
</iq>
```

1. publication d'un élément sur le nœud « table1 »

```
<iq type='get'  
  from='table1@joie.localhost'  
  to='pubsub.joie.localhost'  
  id='items1'>  
<pubsub xmlns='http://jabber.org/protocol/pubsub'>  
  <items node='menu' />  
</pubsub>  
</iq>
```

2. requête du menu par « table1 »

3.3. Index

Figure 1. Schéma de fonctionnement de J.O.I.E 5