

J.O.I.E

# Technical documentation

---

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

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

The goal of this document is to introduce the project JOIE: an ordering system for tactile tablet, mostly for restaurant, can also be used in hotels, bars...

Firstly we will explain the project context, then a presentation, and finally expose the technical aspects: protocols and technologies used. The project revision which matches our document is the 193 of the SVN.

---

## Description of the document

Title	Technical Documention
Date	15 / 11 / 2011
Authors	Antoine MINEAU, Aymeric DERAZEY, Florent LE COZ, Thomas TEIXEIRA, Jacky WONG
E-Mail	<a href="mailto:joie_2012@labeip.epitech.eu">joie_2012@labeip.epitech.eu</a>
Subject	Documentation for developpers
Version	2.7.1

## Revisions

Date	Version	Section	Comment
04/01/11	1		First version
04/07/11	2		Add doxygen documentation links, versioning, check orthography
21/04/11	2.1	1, 2	Add 1&2 sections
28/04/11	2.2	2.1, 3.1	Add schema description, 3.1 section, schema modification
30/04/11	2.3		Add a comparative chart about existing products, add details about the database, Remove interface parts, check orthography, adding the introduction
09/06/11	2.4	2.2, 3.1	Add a comparative chart about existing products, add details about the database, Remove interface parts, check orthography, adding the introduction

15/07/11	2.5	1.2	Remove PSGSXS &Prosody explanation, add Openfire explanation
14/09/2011	2.6		Template edited
15/11/2011	2.6.1		Few changes

## Contents

1.	Project context .....	1
1.1.	Reminder.....	1
1.1.1.	EIP goal .....	1
1.1.2.	Project organization .....	1
1.2.	Environment.....	2
1.2.1.	Realization .....	2
1.2.2.	Technologies.....	2
1.2.3.	Materials .....	3
1.2.4.	Constraints .....	3
2.	Project presentation.....	3
2.1.	Basic principals.....	3
2.2.	Existing on the market .....	5
2.3.	JOIE .....	5
2.4.	Es.....	5
2.5.	No .....	5
3.	Technical presentation.....	5
3.1.	Database description.....	6
3.2.	Index.....	8

---

## 1. Project context

### 1.1. Reminder

#### 1.1.1. EIP goal

One of the fundamental steps during our Epitech degree course to form students to become real professionals is to realize a project with at least 5 students during a period of 18 months called EIP (Epitech innovative project). The EIP gives high experience in project management and an insight of professional life.

In our case, the goal of our EIP is to develop a solution easing the order management in restaurants... Multiple views will be available : client side (display the menu, order, call a server...) , waiter-side (invoicing...), administrator side (account managements, stats...).

#### 1.1.2. Project organization

Our group is divided with :

- network part : le-coz\_f, mineau\_a and teixei\_t
- graphic part : deraze\_a and wong\_j
- documentation, website : mineau\_a

## 1.2. Environment

### 1.2.1. Realization

In order to do in good conditions this project, we gonna use the versioning software Mercurial with CodingTeam, it gives a nice view of our deposit online and various tools to see the progress, bugs evolution, patch... Furthermore, several of us have already used Mercurial during our internship and during the development of school projects.

Depending of the operating system, we will use for the programming Gedit /Emacs/Visual Studio, and also Android's emulator.

### 1.2.2. Technologies

We will only use Python (multi-platform language) and some libraries:

- SleekXMPP is a Python library mostly focused on easing the use of every XMPP extensions as plugins. We use the XEP-006(Publish-Subscribe). The XMPP protocol allows to create nodes in a pubsub server and publish information on these nodes; an event is sent to every entities which has subscribed to this node. Thus, when a client orders something at a table, it is sent to every waiter who are dealing with this table. If the admin decide to update the menu, the server sends the new menu to every client interface. Noe manipulation is required for this update.

- Openfire is a XMPP server written in Java with a web interface numerous extensions are available like Pubsub.

### 1.2.3. Materials

In order to do in good condition this project, we will test it on various touchscreen and also a server who will host our program.

We can also imagine others tools like touch-table or even touch terminals. However at the moment, only the touchscreen is used, this one is the closest to a traditional menu.

### 1.2.4. Constraints

GTK is not the optimal tool in order to get a ergonomic and pleasant interface.

Touchscreens actually sold are not the ideal for the restoration. There is the battery and recharge issue, and also the size which modifies a lot the application readability.

## 2. Project presentation

### 2.1. Basic principals

JOIE's system is a Client / Server. The Server will be the link between the interfaces and the database.

The client part is divided in 3 interfaces using python/GTK, one for each individual :

the customer, used to display the menu, ordering...

the waiter, used to receive the order...

the admin, display statistics, add/modify/delete dish, accounts, tables...

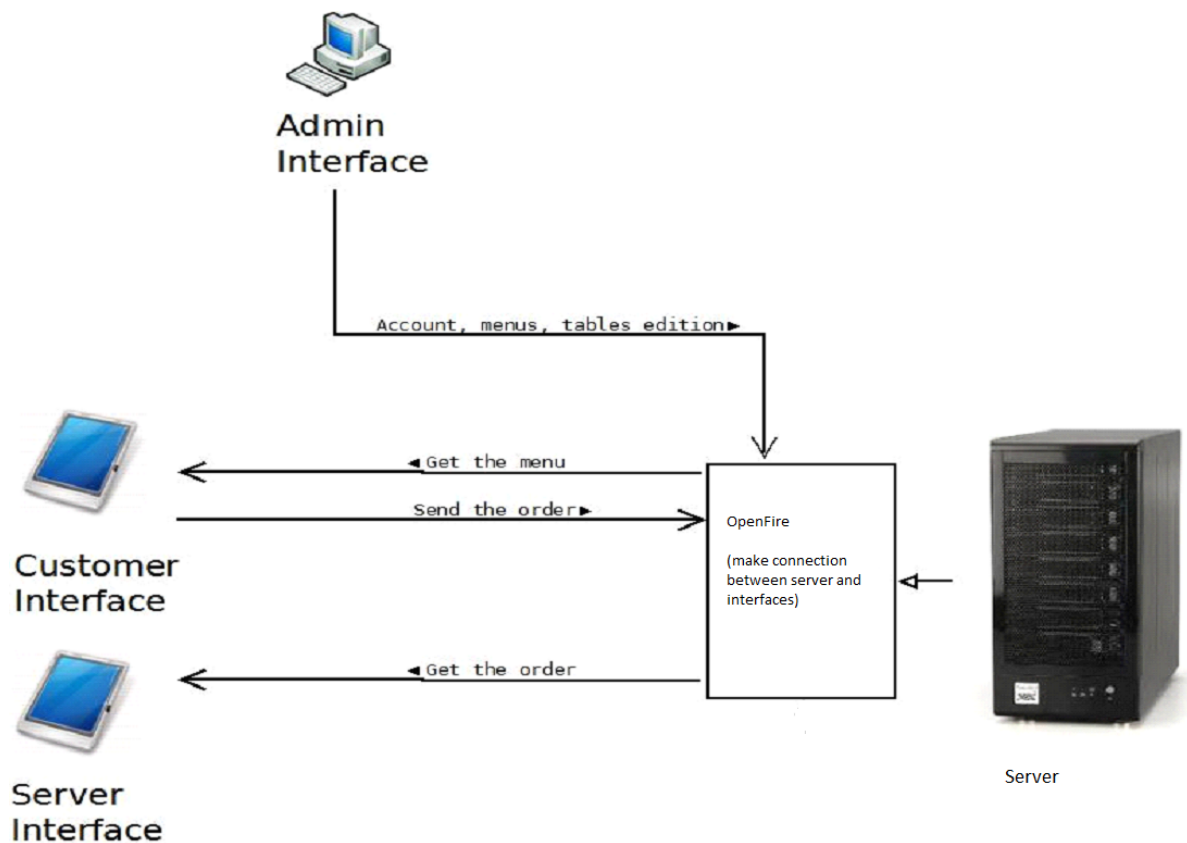


Figure 1. Schéma of functioning

The deployment and utilization of JOIE is simple:

- The admin create the menu, tables, and accounts for his waiters. He must add each waiter to specific tables to receive the events of the table (order, ask for bill...)
- The waiter gives a customer interface
- When the customer orders, the waiter assigned to the table will receive the order details on his interface and can print it for the kitchen.



•When customers want to pay, the waiter can print a global ticket with all the order of the table.

## 2.2. Existing on the market

Similar projects exist on the market :

Functionality	Borne KFC	TChooZ	e-tab	2.3. JOIE
Mobility	No	No	Yes	2.4. Es
Can modify	Yes	No	Yes	2.5. No
Easy to order	Yes	Yes	Yes	Yes
More functionality as well as the order	Take and edit photos	Vote, Messages to other tables	Divide the bill, personalize the command, send the bill by mail	Statistics for the admin
Market aimed	Fast food	Bar	Restaurant	Restaurant, bar

## 3. Technical presentation

The technical documentation is available on:  
<http://eip.epitech.eu/2012/joie/doxygen/index.html>

### 3.1. Database description

Our database is stocked bu openfire :

#### Accounts:

The account are stocked by Prosody, one file by user , this file will take the name of the user (ex : admin.json, user1.json) , and in it we have :

- the encrypted password.

#### Tables:

Each table has a json file with :

- items : orders taken

#### Orders:

Each order has the following fields:

- id
- content : ordered dishes, quantities
- date

Example :

```
'270f1904876ceafb2af95c78d64b21afcdffc49b':  
  { content:<command xmlns='xmpp:joie:ns:command'><dishes  
xmlns='xmpp:joie:ns:command'><dish xmlns='xmpp:joie:ns:command'  
item_id='a9042917abe6566db17cfbde012cb43a3bd13c85'  
quantity='4'></dishes></command>  
  , date: '2011-05-12T13:45:03Z'  
  }
```

### Menu :

The menu is made up with each available dish, a dish has the following fields :

- id
- content: information (name, price, categories, commentary, components)
- encoded image
- date

Example of the field content of a dish :

```
<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>
    ...
    [zipped then base64 encoded jpeg OR png picture]
```

```
...  
</img>  
</dish>
```

For the different customers, developed in Python, we use the library SleekXMPP, and the plugin (xep\_0060.py) allowing the PubSub management.

Some stanza examples of the pubsub protocol :

```
<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>
```

*publishing an item on the node 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>
```

*menu request by table1*

## 3.2. Index

Figure 1.Schéma of functioning..... 4