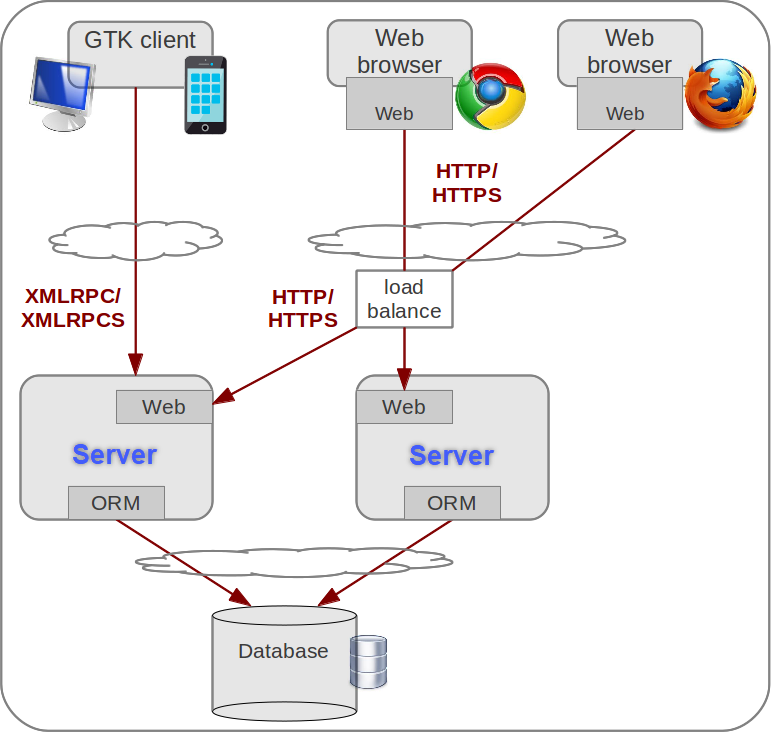
**TÀI LIỆU MÔ TẢ KỸ THUẬT KẾT NỐI MOBILE APP LẤY DỮ LIỆU TỪ SERVER ODOO v11 WSGI**

1. **Phương án kỹ thuật:**

* Odoo là web application dạng wsgi.
* Sử dụng xmlrpc protocol để lấy dữ liệu từ server hiển thị lên Mobile app.
* Kiến trúc hệ thống:



1. **Đặc tả code chi tiết lấy dữ liệu từ server:**

* Link đặc tả External API của server (dùng odoo framework). Trong link này nó có hướng dẫn gọi xmlrpc bằng cả PHP, Ruby, Java, Python: <https://www.odoo.com/documentation/11.0/webservices/odoo.html>
* Ví dụ lấy dữ liệu bằng xmlrpc bằng các ngôn ngữ lập trình khác nhau (code này anh có gửi email rồi nhé)
  + Endpoitn url: <http://103.94.16.226:8069>
  + Database\_name: plasmadb
  + Account: admin
  + Pass: 1@

**Python (python 2.7)**:

|  |
| --- |
| **Code:** |
| import xmlrpclib  db\_name = "mes"  user\_name = 'admin'  pwd = '1@'  # url = 'http://localhost:8080'  url = 'http://103.94.16.226:8069'  extend\_common = '/xmlrpc/2/common'  extend\_object = '/xmlrpc/2/object'  sock\_common = xmlrpclib.ServerProxy(url + extend\_common)  # sock = xmlrpclib.ServerProxy(server\_host + extend\_object)  sock = xmlrpclib.ServerProxy('{}/xmlrpc/2/object'.format(url))  print(sock\_common.version())  uid = sock\_common.authenticate(db\_name, user\_name, pwd, {})  print(uid)  read\_permission = sock.execute\_kw(db\_name, uid, pwd, 'res.partner', 'check\_access\_rights', ['read'],  {'raise\_exception': False})  print(read\_permission)  ids = sock.execute\_kw(db\_name, uid, pwd, 'res.partner', 'search', [[]], {'limit': 3})  print(ids)  # record\_id = sock.execute\_kw(db\_name, uid, pwd, 'res.partner', 'read', [ids], {})  record\_id = sock.execute\_kw(db\_name, uid, pwd, 'res.partner', 'read', [ids],  {'fields': ['name', 'country\_id', 'comment']})  print '# Return record: ', record\_id |
| **Kết quả:** |
|  |

* + **Java**:

|  |
| --- |
| **Code:** |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package odoorpc;  import java.net.MalformedURLException;  import java.net.URL;  import static java.util.Arrays.asList;  import static java.util.Collections.emptyList;  import static java.util.Collections.emptyMap;  import java.util.HashMap;  import java.util.List;  import org.apache.xmlrpc.XmlRpcException;  import org.apache.xmlrpc.client.XmlRpcClient;  import org.apache.xmlrpc.client.XmlRpcClientConfigImpl;  /\*\*  \*  \* @author Admin  \*/  public class OdooRPC {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) throws MalformedURLException, XmlRpcException {  final String url = "http://103.94.16.226:8069",  db = "plasmadb",  username = "admin",  password = "1@";  final XmlRpcClient client = new XmlRpcClient();  final XmlRpcClientConfigImpl common\_config = new XmlRpcClientConfigImpl();  common\_config.setServerURL(  new URL(String.format("%s/xmlrpc/2/common", url)));  Object obj = client.execute(common\_config, "version", emptyList());  int uid = (int) client.execute(  common\_config, "authenticate", asList(  db, username, password, emptyMap()));  final XmlRpcClient models = new XmlRpcClient() {  {  setConfig(new XmlRpcClientConfigImpl() {  {  setServerURL(new URL(String.format("%s/xmlrpc/2/object", url)));  }  });  }  };  Object readPermission = models.execute("execute\_kw", asList(  db, uid, password,  "res.partner", "check\_access\_rights",  asList("read"),  new HashMap() {  {  put("raise\_exception", false);  }  }  ));  List<Object> l = asList((Object[]) models.execute("execute\_kw", asList(  db, uid, password,  "res.partner", "search\_read",  asList(asList()),  new HashMap() {  {  put("fields", asList("name", "country\_id", "comment"));  put("limit", 5);  }  }  )));    System.out.println(obj.toString() + readPermission);  System.out.println("# User Id:" + uid);  System.out.println("# Return:" + l);  }  } |
| **Kết quả:** |
|  |

1. **Đặc tả Model:**

* Model: *res.partner*

|  |  |
| --- | --- |
| **Model name** | res.partner |
| **Fields dữ liệu** | *id (int)*  *name (string)*  *country\_id*  *comment* |
| **Mô tả** | Model thông tin về Profile User, danh sách User của hệ thống.   * Model lấy dữ liệu: *res.partner* * Điều kiện lấy dữ liệu: *is\_company = false và active = true và customer = true* |

* Model: *p.customer*

|  |  |
| --- | --- |
| **Model name** | **p.customer** |
| **Fields dữ liệu** | *Id*  *name*  *address* |
| **Mô tả** | Model thông tin về khách hàng. |

* Model: *p.equipment*

|  |  |
| --- | --- |
| **Model name** | **p.equipment** |
| **Fields dữ liệu** | *Id*  *code: <mã thiết bị>*  *stage: <trạng thái>*  *warehouse: <kho>*  *p\_customer: <khách hàng đang sử dụng>*  *description: <thông số kỹ thuật>* |
| **Mô tả** | Model thông tin về thiết bị.  +) Trạng thái: STAGE = [(0, **'Không xác định'**), (1, **'Vỏ'**), (2, **'Tái nạp'**), (3, **'Bình tồn'**), (4, **'Bình đang sử dụng'**)]  +) Kho: WAREHOUSE = [(0, **'Không xác định'**), (1, **'Kho Công ty'**), (2, **'Kho Nhà máy'**), (3, **'Kho Khách hàng'**)]  +) Khách hàng đang sử dụng: Lấy dữ liệu từ model p.customer |

1. Mô tả nghiệp vụ khách hàng yêu cầu:

* **Bước 1: Quét thiết bị để tách mã**.
  + Mô tả nghiệp vụ: Ấn vào quét và dùng Cam của điện thoại để quét các thiết bị. Đây là một quá trình liền mạch. Ví dụ, mỗi 1 lần có xe đưa thiết bị về khoảng 50 – 200 thiết bị, nhân viên sẽ dùng điện thoại quét mã nhiều bình do đó cần liền mạch và hạn chế thao tác bấm nốt, thao tác chờ hết sức có thể (Hiện tại bản demo đang phải bấm nhiều lần, chờ lâu mới thao tác xong đối với 1 thiết bị 🡪 em xem giải pháp cải thiện cái chỗ bấm nhiều lần nhé).
  + Mong muốn: Ứng dụng tự động chụp ảnh sau 1 – 3 giây rồi nhận diện và chuyển trạng thái thiết bị chứ không phải ấn chụp, gửi ảnh (làm tương tự như quét mã QR code). Quy tắc lấy ra chuỗi mã đó là:
    - Lấy ra chuỗi tối đa 7 ký tự.
    - Bắt đầu bằng chữ và kết thúc bằng số cuối cùng (lý do là tồn tại 2 loại thiết bị có 6 và 7 mã).
    - *Ví dụ: Khi quét ra mã là HH05059TR 🡪 cắt ra là HH05059 (tức là bắt đầu bằng chữ và kết thúc bằng 1 số trước khi lại xuất hiện chữ tiếp); khi quét ra mã là DH0304TR225 thì cắt ra là DH0304.*
  + Một số comment của khách hàng sau buổi demo:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

* **Bước 2: Tạo ra Đơn hàng tương ứng để chuyển trạng thái của thiết bị**.
  + Mô tả nghiệp vụ: Khi quét xong phải chuyển trạng thái của Thiết bị sang next step (theo file .pdf hôm trước anh gửi cho em về các trạng thái và hành động của vòng đời thiết bị). Vậy mình phải xử lý 2 việc, chuyển trạng thái thiết bị và tạo 1 đơn hàng để lưu lại lịch sử thời điểm chuyển trạng thái thiết bị. Về giải pháp món này, em có thể xem là mình quét xong hết 1 lượt khi nào dừng thì mới tạo đơn hàng và chuyển toàn bộ trạng thái thiết bị trong Database có được ko? Kiểu batch update ấy em nhé.
  + Hiện trạng: Chưa có
  + Mong muốn: Hoàn thành càng sớm càng tốt.
  + Khách hàng comment sau demo:

|  |
| --- |
|  |