Software Requirements Specification

Improvement for Lunch Order Module

Prepared by: **Thu Nguyen**

Jirakit Paitoonnaramit

September, 2017

Revision History

Date	Version	Description	Author
20/09	1.0	Draft version	Thu Nguyen
29/09	1.1	Complete document	Thu Nguyen
30/09	1.1	Update after reviewing	Thu Nguyen, Jirakit Paitoonnaramit

Table of Contents

1 Introduction 5
1.1 Purpose 5
1.2 Background and Scope 5
1.3 Definitions, acronyms, and abbreviations 5
1.4 References 6
1.5 Overview 6
2 Overall description 7
2.1 Product perspective 7
2.1.1 Brief introduction about the existing module 7
2.2 Production function 8
2.3 User 9
2.4 Constraints 9
3 Specific requirements 10
3.1 Functional requirements 10
3.1.1 User Class 1 - The LunchOrderLineSendEmail 10
3.1.2 User class 2: FavoriteOrderLine 10
3.2 Non functional requirement 10
3.2.1 Reliability 11
3.2.2 Usability 11
3.2.3 Maintainability 11
3.2.4 Security 11

List of figures

Figure 1 Architecture of Odoo..... 7
Figure 2 The function 'I am feeling luck' 8
Figure 3: The MVC design pattern of Odoo..... 9

1 Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the "Improvement for Lunch Order Module" project. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for development team including designers, developers and testers to develop the first version of the system for the development team.

1.2 Background and Scope

Many companies order sandwiches, pizzas and other, from usual suppliers, for their employees to offer them more facilities. However lunch management within the company requires proper administration especially when the number of employees or suppliers is important.

The "Lunch Order" module has been developed to not only make this management easier but also offer employees more tools and usability. However, the existing version of this module doesn't support to send emails to vendors and the feature Favorite Order is not good enough which is only based on the last order. The "Lunch Order" module has been developed to not only make this management easier but also offer employees more tools and usability. However, the existing version of this module doesn't support to send emails to vendors and the feature Favorite Order.

This project is to extend the "Lunch Order" module, providing the feature "Sending email to vendors" and improve the feature Favorite Order.

1.3 Definitions, acronyms, and abbreviations

No.	Term	Definition
1	User	Someone who interacts with the mobile phone application
2	Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
3	SRS	Software requirement specification
4	FR	Functional requirement
5	DEP	Dependency
6	"Confirmed" state	An order is in "Confirmed" state when all order lines of that order are confirmed

Table 1: Definition

1.4 References

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

1.5 Overview

The rest of the SRS is organized by parts

1. Overall description

Describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in section 3

2. Specific Requirement

This section contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems.

2 Overall description

2.1 Product perspective

This module will be based on Odoo which offers a range of business applications, forming a complete suite of enterprise management applications.

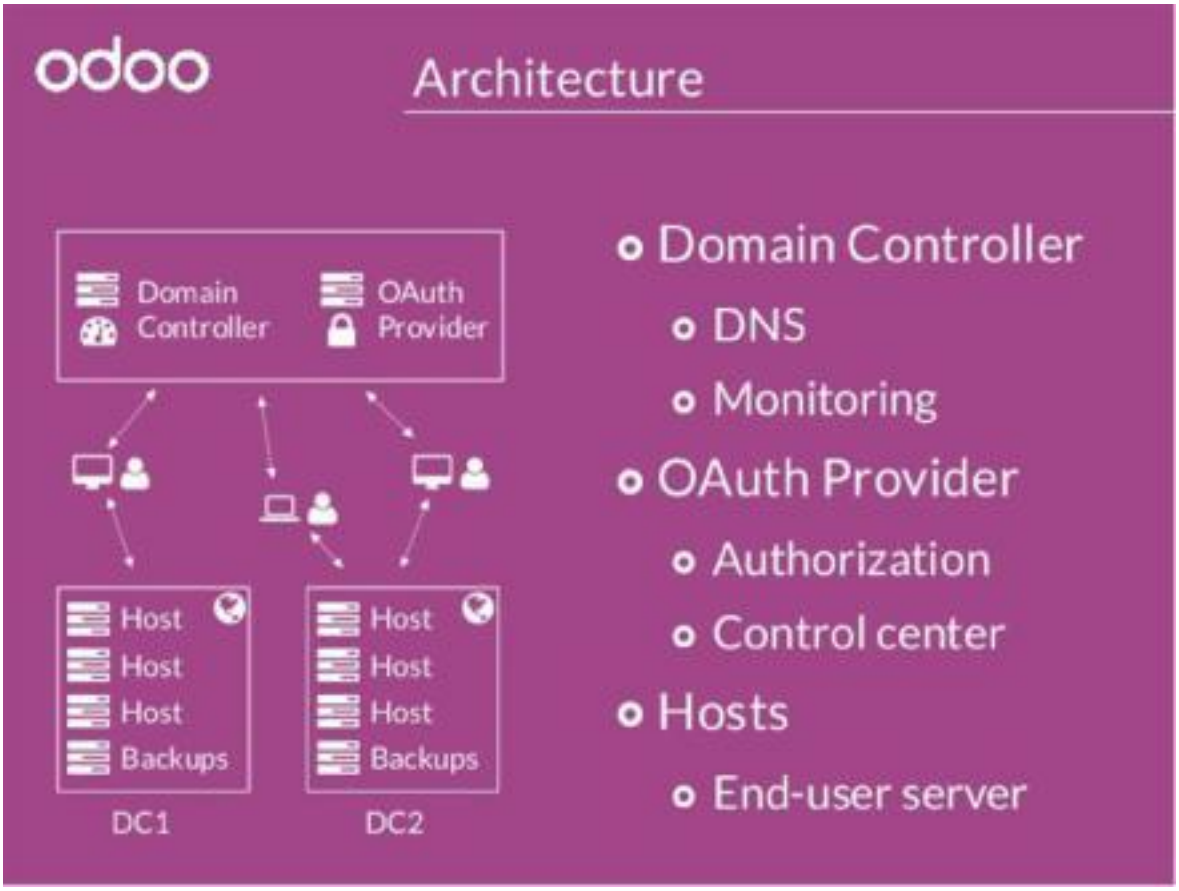

Figure 1 Architecture of Odoo

2.1.1 Brief introduction about the existing module

Currently, the existing version of Lunch order provides following features:

1. Create an order
   1.1 It allows the user to select the last order from each vendor.
2. Products:
   2.1 Create, import, delete, archive products.
3. Product categories:
   3.1 Create, import, delete product category
4. Alerts
   4.1 Create, import, and delete alerts

5. Today's Orders

   5.1 View, cancel Orders

6. Orders by Vendor

   6.1 View, cancel Orders

7. Control Accounts

   7.1 Create, Delete, Import, Export Control Accounts

8. Employee Payments

   8.1 Create, delete, import, export Employee Payments

9. Previous Orders

   9.1 View, cancel previous orders

10. Your Lunch Accounts

    10.1 Delete, Export the information of account

The existing module does ***not*** support to

- Send email about orders to vendors
- Make favorite orders, but a similar existing feature, called "I'm feeling lucky" which is currently based on the last order by vendors

Figure 2 The function 'I am feeling luck'

2.2 Production function

This project will develop 2 new features for the module "Lunch Order":

- Sending email to vendors
  - This will send an email of all orders created on that day. This email will send at 10h30 everyday.

- Create favorite order
  - This feature allows users to create a quick order from a list of favorite order lines.

2.3 User

The potential users are employees of organizations and companies who are supposed to be familiar and have various experiences of using web-applications. They may have no technical knowledge.

2.4 Constraints

The software is based on Odoo, therefore the software has to follow the MVC design pattern as other modules of Odoo.

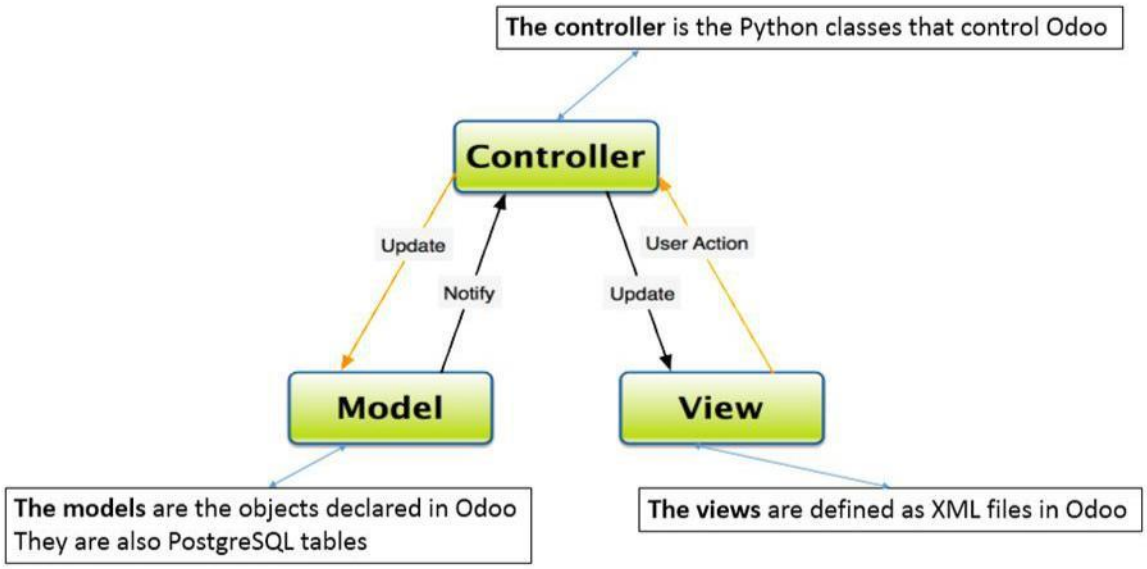

Figure 3 The MVC design pattern of Odoo

3 Specific requirements

3.1 Functional requirements

3.1.1 User Class 1 - The LunchOrderLineSendEmail

3.1.1.1 Functional requirement 1.1

ID: FR1
TITLE: Send email to a vendor.

DESC: The system will send an email when the user confirms the lunch order line. The email includes only the information of only that order line about product name, price, notes, and company name of the user and address of the user.
If the price, note and address are not available, the system will not include them in the email.
DEP: None

ID: FR2
TITLE: Not send email to a vendor if it doesn't have email address.
DESC: If the user doesn't input the email address of vendors, the system will not send email to that vendor.
DEP: FR1

ID: FR3
TITLE: Only user in Manager Group can send email to vendors.
DESC: Only user in Manager Group can send email to vendors. Normal users cannot access this feature.
DEP: FR1

3.1.2 User class 2: FavoriteOrderLine

3.1.2.1 Functional requirement 1.2

ID: FR4
TITLE: Make a list of favorite order lines
DESC: The system shall create a list of favorite order lines, which includes the three most frequent order lines of each user. Each order line is equivalent to one product of one vendor. The most frequent order line is an order line that the number of times that the user chooses the product from one vendor is the most.
DEP: None

ID: FR5
TITLE: Add order lines to lunch order
DESC: When a user selects order lines from the list of favorite order line, the system shall add order lines to the lunch order.
DEP: FR4

3.2 Non functional requirement

3.2.1 Reliability

The system should not crash during operation before 10h30 everyday.

3.2.2 Usability

User Interface elements should be easy to understand.

The features should be systematic and consistent so the user could learn easily.

3.2.3 Maintainability

The features should be expendable. So that they could be extended in the future when the customer's needs change.

3.2.4 Security

The features should be secure. Only the authorized users can access the features and execute them.