****

**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Car Rental Portal that manages the business of a bakery, only product for schools**

|  |  |
| --- | --- |
| **Group 1** | |
| **Group members** | Trần Hữu Đức – SE61448  Huỳnh Công Thành – SE61297  Lê Vũ Đăng Khoa – SE61238  Nguyễn Tường Tâm – SE61384 |
| **Supervisor** | Lâm Hữu Khánh Phương |
| **Ext. Supervisor** | N/A |
| **Capstone Project Code** | CRP |

- Ho Chi Minh City, 01 September 2016 -

This page is intentionally left blank

Table of Contents

[Table of Contents 3](#_Toc437062951)

[List of Tables 7](#_Toc437062952)

[List of Figures 9](#_Toc437062953)

[Definitions, Acronyms, and Abbreviations 13](#_Toc437062954)

[A. Introduction 14](#_Toc437062955)

[1. Project Information 14](#_Toc437062956)

[2. Introduction 14](#_Toc437062957)

[3. Current Situation 14](#_Toc437062958)

[4. Problem Definition 14](#_Toc437062959)

[5. Proposed Solution 15](#_Toc437062960)

[5.1. Feature functions 15](#_Toc437062961)

[5.2. Advantages and disadvantages 15](#_Toc437062962)

[6. Functional Requirements 15](#_Toc437062963)

[7. Roles and Responsibility 16](#_Toc437062964)

[B. Software Project Management Plan 17](#_Toc437062965)

[1. Problem Definition 17](#_Toc437062966)

[1.1. Name of this Capstone Project 17](#_Toc437062967)

[1.2. Problem Abstract 17](#_Toc437062968)

[1.3. Project Overview 17](#_Toc437062969)

[2. Project organization 20](#_Toc437062970)

[2.1. Software Process Model 20](#_Toc437062971)

[2.2. Roles and responsibilities 21](#_Toc437062972)

[2.3. Tools and Techniques 22](#_Toc437062973)

[3. Project Management Plan 22](#_Toc437062974)

[3.1. Tasks 22](#_Toc437062975)

[3.2. Task sheet 24](#_Toc437062976)

[3.3. All Meeting Minutes 24](#_Toc437062977)

[4. Coding Convention 25](#_Toc437062978)

[C. Software Requirement Specification 26](#_Toc437062979)

[1. User Requirement Specification 26](#_Toc437062980)

[1.1. Guest requirement 26](#_Toc437062981)

[1.2. Customer requirement 26](#_Toc437062982)

[1.3. Staff requirement 26](#_Toc437062983)

[1.4. Manager requirement 27](#_Toc437062984)

[2. System Requirement Specification 28](#_Toc437062985)

[2.1. External Interface Requirement 28](#_Toc437062986)

[2.2. System Overview Use Case 28](#_Toc437062987)

[2.3. List of Use Case 30](#_Toc437062988)

[3. Software System Attribute 128](#_Toc437062989)

[3.1. Usability 128](#_Toc437062990)

[3.2. Reliability 129](#_Toc437062991)

[3.3. Availability 129](#_Toc437062992)

[3.4. Security 129](#_Toc437062993)

[3.5. Maintainability 129](#_Toc437062994)

[3.6. Portability 129](#_Toc437062995)

[3.7. Performance 129](#_Toc437062996)

[4. Conceptual Diagram 130](#_Toc437062997)

[D. Software Design Description 132](#_Toc437062998)

[1. Design Overview 132](#_Toc437062999)

[2. System Architecture Design 133](#_Toc437063000)

[2.1. Web Application architecture description 133](#_Toc437063001)

[3. Component Diagram 134](#_Toc437063002)

[4. Detailed Description 135](#_Toc437063003)

[4.1. Class Diagram 135](#_Toc437063004)

[4.2. Class Diagram Explanation 138](#_Toc437063005)

[4.3. Interactive Diagram 150](#_Toc437063006)

[5. User Interface Design 207](#_Toc437063007)

[5.1 User Login 207](#_Toc437063008)

[5.2 Order A Product 208](#_Toc437063009)

[5.3 Add to cart a product 208](#_Toc437063010)

[5.4 Check cart 209](#_Toc437063011)

[5.5 View information in cart 210](#_Toc437063012)

[5.6 Confirm order 211](#_Toc437063013)

[5.7 Add customer information to order 211](#_Toc437063014)

[5.8 View order list 212](#_Toc437063015)

[5.9 Staff add product to order 213](#_Toc437063016)

[5.10 Add customer to order 214](#_Toc437063017)

[5.11 Check out with new customer 215](#_Toc437063018)

[5.12 View input bill list 216](#_Toc437063019)

[5.13 Add new input bill 217](#_Toc437063020)

[5.14 View input material list 218](#_Toc437063021)

[5.15 View input material detail 219](#_Toc437063022)

[5.16 Add new input material 220](#_Toc437063023)

[5.17 Review income by time 221](#_Toc437063024)

[5.18 Review income weekly detail 223](#_Toc437063025)

[5.19 VAT Tax Declaration – Appendix 1 224](#_Toc437063026)

[6. Database Design 226](#_Toc437063027)

[6.1 Entity relationship diagram 226](#_Toc437063028)

[6.2 Entity Dictionary 228](#_Toc437063029)

[7. Algorithms 240](#_Toc437063030)

[7.1 Order State 240](#_Toc437063031)

[E. 番５のレポート。システムの展開とテスト 242](#_Toc437063032)

[1. はじめに 242](#_Toc437063033)

[1.1 概要 242](#_Toc437063034)

[1.2 テストアプローチ 242](#_Toc437063035)

[2. データベース関係ダイアグラム 242](#_Toc437063036)

[2.1 ウェブアプリケーション物理的なダイアグラム 242](#_Toc437063037)

[2.2 ウェブアプリケーションデータ辞書 244](#_Toc437063038)

[3. パフォーマンス対策 261](#_Toc437063039)

[3.1 製品検索、メンバーのパフォーマンス 261](#_Toc437063040)

[3.2 送り状リストが展示されるパフォーマンス 261](#_Toc437063041)

[3.3 収入統計など展示されるパフォーマンス 261](#_Toc437063042)

[4. テスト計画 261](#_Toc437063043)

[4.1 テストする機能 261](#_Toc437063044)

[4.2 テストされない機能 262](#_Toc437063045)

[5. システムテストのテストケース 262](#_Toc437063046)

[5.1 ウェブシステムのテストケース 262](#_Toc437063047)

[F. 番６のレポート。ソフトウェアユーザーズマニュアル 304](#_Toc437063048)

[1. インストレーションガイド 304](#_Toc437063049)

[1.1. サーパ側の環境を設定 304](#_Toc437063050)

[1.2. サーパ側で展開 304](#_Toc437063051)

[1.3. クライアント側の環境の設定 308](#_Toc437063052)

[2. ユーザーガイド 309](#_Toc437063053)

[2.1. ログイン 309](#_Toc437063054)

[2.2. 製品を注文すること 310](#_Toc437063055)

[2.3. カートに製品を追加すること 310](#_Toc437063056)

[2.4. かーとをチェックすること 311](#_Toc437063057)

[2.5. カーとの精細 312](#_Toc437063058)

[2.6. 送り状を確認すること 313](#_Toc437063059)

[2.7. 送り状に顧客の情報を充実すること 313](#_Toc437063060)

[2.8. 送り状のリストを見ること 314](#_Toc437063061)

[2.9. 「人員」送り状を造ること 315](#_Toc437063062)

[2.10. 送り状に顧客を追加すること 316](#_Toc437063063)

[2.11. 新い顧客の口座で送り状をチェックアウト 317](#_Toc437063064)

[2.12. 入力法案 リスク 318](#_Toc437063065)

[2.13. 新入力法案を造ること 319](#_Toc437063066)

[2.14. 入力原料 リスクを見る 320](#_Toc437063067)

[2.15. 入力原料 の精細を見ること 321](#_Toc437063068)

[2.16. 新入力原料を造ること 322](#_Toc437063069)

[G. Appendix 324](#_Toc437063070)

List of Tables

List of Figures

Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| CRP | Car Rental Portal |

Table 1: Definitions, Acronyms, and Abbreviations

# Introduction

## Project Information

* Project name: **Car Rental Portal**
* Project Code: **CRP**
* Product Type: **Website Application**
* Start Date: **September 1st, 2016**
* End Date: **December 1st, 2016**

## Introduction

In this document, we introduce a solution for online car rental service.

Our solution is a website application that help small and large car rental providers to promote their service online and customer to easily find and book the most suitable car for their trip. The solution allows providers to define their own rental policy and price. It also simplifies the providers’ rental management process.

## Current Situation

Nowadays, there are many car rental providers in Vietnam, however, only a small percent of them have utilized internet to promote their business. Part of them (Large rental companies, taxi companies) have their own website or mobile application, while some choose to become driver for online rental portal solution like Uber and Grab.

Many small and middle-sized providers cannot afford their own website or mobile application, and also find contract with Uber and Grab too rigid because they are not allowed to define their own rental price and policy. Uber’s and Grab’s solutions also do not apply well for long-period rental.

A new solution that has just appeared recently is Aleka. They allows providers to post their vehicles on their website, helps customer to find the most suitable vehicle in their system, and then help arranging the booking between customer and provider. Their solution however does not offer any way for customer to rate and comment about the service of provider, as well as a way for customer to read others’ ratings and reviews before deciding to book the vehicle.

## Problem Definition

The current situation has several problems:

* **Cost barrier:** The cost to develop an online rental solution is too high for small rental providers.
* **Incompatible rental portal solutions:** There are great rental portal services like Uber, Grab, or Aleka on the market. However, their solutions have disadvantage, such as limiting provider’s right, cannot handle long-period rental very well, or lack of a mean for rating and reviewing provider’s service.

## Proposed Solution

Our proposed solution, named Car Rental Portal, is a car rental portal website application that can provide a flexible mean for provider to promote their rental service online while still allows them to define their own rental policy. Our solution also allows customer to rate and comment about their past renting, as well as read others’ review on vehicles. Furthemore, with our vehicle searching function, which integrate a recommendation engine inside, customers can easily find the best vehicle that suit their need.

### Featured functions

* **Manage online rental service:** Provider can manage their vehicles’ and bookings’ information, as well as declare their own rental policies, including price contrainst, time constraint, travel distance constraint…
* **Advanced vehicle searcher with recommender engine:** System can help customers to find the most suitable vehicle fast and easily based on their booking history.
* **Review vehicle and read vehicle’s reviews:** Customer can comment and rate rental service after renting vehicle, as well as read reviews from other customer for a vehicle.
* **Book vehicle and cancel booking:** Customer can book and pay booking fee through NganLuong payment portal, as well as cancel their booking easily.

### Advantages

* Offer a cheap and easy online rental management solution.
* Allow provider to define their own rental policies.
* Offer vehicle searching with recommender engine.
* Offer rental service reviewing function.
* Allow paying through NganLuong payment portal.

### Disadvantages:

* Since recommender works depending on customer’s booking history, customer must have booked at least once in the past to utilize it.
* Depends on NganLuong and has no alternative payment method in case NganLuong goes down.

## Functional Requirements

* Manage vehicles:
  + Provider can create, update and delete vehicle.
* Manage vehicles’ bookings:
  + Provider can keep track of their vehicle’s booking schedule.
  + Provider can create self-booking to reserve their vehicle.
  + Provider can cancel self-booking.
* Manage garage:
* Manage price group:
* Manage booking
  + Customer can book vehicle
  + Customer can cancle booking
  + Customer can keep track of their booking history
  + Customer can rate and comment about their booking.
* Search and review vehicle

## Roles and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Lâm Hữu Khánh Phương | Project Manager | Supervisor | phuonglhk@fpt.edu.vn |
| 2 | Trần Hữu Đức | Developer | Leader | ducthse61448@fpt.edu.vn |
| 3 | Huỳnh Công Thành | Developer | Member | thanhchse61297@fpt.edu.vn |
| 4 | Lê Vũ Đăng Khoa | Developer | Member | khoalvdse61238@fpt.edu.vn |
| 5 | Nguyễn Tường Tâm | Developer | Member | tamntse61384@fpt.edu.vn |

Table 2: Roles and Responsibility

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name**: Bakery Management Application that manages the business of a bakery, only product for schools
* **Vietnamese name**: Ứng dụng quản lý cửa hàng cung cấp bánh ngọt cho các trường học
* **Abbreviation**: BMA

### Problem Abstract

Nowadays when the canteens or food holes in the schools want to order the bakery the high quantity of cakes, for the first time, they often go to the bakery to choose the cakes, order with the quantity. For the next time, customer can order directly or by the phone. On the other hand, when the manager of the bakery wants to control the business; work out the total of revenue, tax manually.

We provide a web application to help the customer’s order the cake with high quantity, track the order easily. Moreover the web application also helps the bakery manage the customer, product, input material, order. In addition, this application also helps the bakery manage input and output bills; review the income by time, by type of goods, the revenue per customer and review VAT tax monthly and excise tax yearly.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* **Server crash:** The web application runs on the server. So if server crash, the web application cannot run.
* **Absence of team members:** team members can get sick or unexpected problems.
* **Shortage of technology knowledge**

#### The Proposed System

According the requirement of the bakery manager, we build a web application, which can provide the way for the customer’s order the product online and the bakery can control the business easily.

##### Website

* For guests:
  + Guests can view and search products in store.
  + Guests can order the products online.
  + Guests will receive an account from staff when the first guest’s order is approved.
* For customers:
  + Customers can order the products online.
  + Customers can view their order state.
  + Customers can edit/cancel their orders.
  + Customers can edit their information,
  + Customers can confirm order when the order is edited by staff.
* For staffs:
  + Staffs can manage products (add, edit, change status). In addition, they can add, edit materials to products.
  + Staffs can manage input materials (add, edit), discard input material.
  + Staffs can add bill, edit, export bill.
  + Staffs can create account for customer (when the first customer’s order is approved), change status of customer account.
  + Staffs can change order state, add/edit customer’s order, approve order.
  + Staffs can manage product materials (add, edit, change status).
* For manager:
  + Manager can create VAT, excise tax declaration and edit the tax rate.
  + Manager can view statistic (income by time, product;revenue by customer).
  + Manager can edit store information.
  + Manager can manage staff account (create, change status)
  + Manager can manager customer account (change status, change to loyal)
  + Manager can manager supplier (create, edit, change status)
* Notfications: The system will show the notification when:
  + - To staff:
      * The material is below standard quantity
      * The new order is created
    - To customer:
      * The order is changed state.
      * The order is edited by staff.

#### Boundaries of the System

* Every bakery store that needs to manage their store easier can deploy this system.
* Customer is school’s canteen or food holes.
* The language of this system is Vietnamese
* The complete product includes:
  + Order system for guests and customers.
  + Manage system for staffs and manager.

#### Future Plans

* Support export financial report.
* Provide the financial management.
* Provide the personnel management.
* Provide promotion functions

#### Development Environment

##### Hardware requirement

For web server:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Network Bandwidth | 4 Mbps | 8 Mbps |
| Operating System | Window Server 2008 | Window Server 2008 |
| Computer Processor | Intel® Xeon Dual Core® 1.4GHz (4MB Cache) | Intel® Xeon® Quad Core (8M Cache, 2.40 GHz) |
| Computer Memory | 2GB RAM | 4GB or more |
| Hard disk | 40GB | 80GB |

Table 3: Hardware requirement for Web Server

For web development:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512 Kbps | 8 Mbps |
| Operating System | Window Vista, 7, 8 | Window 7, 8 |
| Computer Processor | 1 GHz | Intel® Core™ i5 CPU 2.53 GHz |
| Computer Memory | 2GB RAM | 4GB or more |

Table 4: Hardware requirement for Web Server

##### Software requirement

|  |  |
| --- | --- |
| Software | Name / Version |
| Operating system | Windows 7 or above |
| Modeling tool | StarUML version 2.5.0 |
| IDE | Microsoft Visual Studio 2013 Ultimate |
| DBMS | SQL Server 2008 R2 Developer |
| Source control | Github 3.0.5 |
| Web browser | Chrome 42 or above |

Table 5 : Software requirement

## Project organization

### Software Process Model

This project is developed under modified waterfall model (Sashimi). We choose this model because the following reasons:

* Due to the timeline of capstone project, this project lasts 4 months, a short project.
* The requirements of the project are well-defined and well-understood by all team members.
* During the process of the capstone project, the team have to submit reports corresponding with the deliverables in each phase.

Requirement

Design and Architecture

Development and Coding

QA and Testing

Maintenance

Figure 1: Modified Waterfall Development Model

For more information about modified waterfall development model: <http://www.waterfall-model.com/sashimi-waterfall-model/>

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Ngô Đăng Hà An | Supervisor / Project Manager | - Clarify user requirement  - Give out technical support and business analysis  - Track development process  - Review document and product |
| 2 | Huỳnh Hữu Trường Giang | Team leader, BA, Developer, Tester | - Track process  - Plan project, distribute tasks  - Requirement analysis  - Design database  - Prepare documents  - Design GUI  - Code  - Test  - Deploy product |
| 3 | Ông Tấn Mạnh | Team Member, BA, Developer, Tester | - Analysis requirements  - Design database  - Prepare documents  - Design GUI  - Code  - Test |
| 4 | Nguyễn Quốc Vương | Team Member, BA, Developer, Tester | - Analysis requirements  - Design database  - Prepare documents  - Design GUI  - Code  - Test |

Table 6: Roles and responsibilities

### Tools and Techniques

|  |  |
| --- | --- |
| Tool / Technique | Name / version |
| Frontend | HTML5, CSS3, JavaScript, jQuery, Bootstrap, SignalR |
| Backend | ASP.NET MVC 5 |
| Web server | IIS 8 |
| Development tool | Microsoft Visual Studio 2013 Ultimate |
| DBMS | MSSQL Server 2008 R2 Developer |
| Source control | Github 3.0.5 |
| Modeling tool | StarUML 2.5.0 |
| Document tool | Microsoft Word 2013 |

Table 7: Tools and Techniques

## Project Management Plan

### Tasks

Below are all the major tasks that need to be performed sequentially during the development of the system.

#### Task 1: Initiating

|  |  |
| --- | --- |
| Task name | Initiating |
| Description | - Get requirements about the business of the bakery.  - Research how to calculate the taxes include VAT and excise tax.  - Offer team’s solution, declare its pros and cons, introduce some core function of the new system. |
| Deliverables | Report No.1 Introduction |
| Resource needed | 20 man-days |
| Dependencies and constraints | N/A |
| Risk | - Missing requirement.  - Project’s scope can be unclear.  - Lack of member share and understand.  - Lack of work force |

#### Task 2: Planning

|  |  |
| --- | --- |
| Task name | Planning |
| Description | - Review team’s solution, verify project’s scope and determine project organization  - Create project management plan |
| Deliverables | Report No.2 Software Project Management Plan |
| Resource needed | 16 man-days |
| Dependencies and constraints | - Base on Report No.1 Introduction.  - Planned project must be completed in 14 weeks |
| Risk | - Lack of experience  - Plan may not fit yet |

#### Task 3: Specifying requirements

|  |  |
| --- | --- |
| Task name | Specifying requirements |
| Description | Identify and clarify software requirements. |
| Deliverables | Report No. 3 Software Requirement Specification |
| Resource needed | 20 man-days |
| Dependencies and constraints | Base on Report No.2 Software Project Management Plan. |
| Risk | - Misunderstood or unclear system’s requirement  - Lack of practical experience |

#### Task 4: Designing database

|  |  |
| --- | --- |
| Task name | Designing database |
| Description | Design database in conceptual, logical and physical |
| Deliverables | Physical database and SQL script. |
| Resource needed | 12 man-days |
| Dependencies and constraints | Base on Report No. 3 Software Requirement Specification |
| Risk | Lack of practical experience leading to unreasonable database design |

#### Task 5: Create Software Design Description

|  |  |
| --- | --- |
| Task name | Create Software Design Description |
| Description | Decide software architect and clarify software detail design. |
| Deliverables | Report No. 4 Software Design Description |
| Resource needed | 40 man-days |
| Dependencies and constraints | Base on Report No. 3 Software Requirement Specification and designed database |
| Risk | Lack of practical experience leading to unreasonable software design |

#### Task 6: Implementing

|  |  |
| --- | --- |
| Task name | Implementing |
| Description | Implements all functions of system. |
| Deliverables | Software package. |
| Resource needed | 80 man-days |
| Dependencies and constraints | - Base on Software Requirement Specification and Software Design Description.  - Coding try to follow coding convention. |
| Risk | - Member does not perform unit test.  - Lack of practical experience |

#### Task 7: Testing

|  |  |
| --- | --- |
| Task name | Testing |
| Description | - Create test plan  - Perform integration test and system test. |
| Deliverables | Report No. 5 System Implementation & Test |
| Resource needed | 40 man-days |
| Dependencies and constraints | Implementation is finished |
| Risk | - Lack of testing experience leading to lack of test cases.  - Not enough time for performing test. |

#### Task 8: Creating User’s Manual

|  |  |
| --- | --- |
| Task name | Creating User’s Manual |
| Description | Create the user’s manuals to instruct users using system. |
| Deliverables | Report No.6 Software User’s Manual |
| Resource needed | 12 man-days |
| Dependencies and constraints | Testing is finished |
| Risk | User’s manual may be difficult for user to understand and confuse. |

### Task sheet

Refer to “Task sheet” folder.

### All Meeting Minutes

Refer to “Meeting minutes” folder.

## Coding Convention

This project follows “C# coding conventions”.

* Naming Convention :
  + - For variable’s name, use camel case. Example: lstProduct, isAdmin…
    - For function name, class name, use pascal case. Example: GetNotifications, SearchProduct…
* Layout Convention :
  + - Use the default Code Editor Settings (smart indenting, four-character indents, tabs saved as spaces).
    - Write only one statement/declaration per line.
    - If continuation lines are not indented automatically, indent them one tab stop (four spaces).
    - Add at least one blank line between method definitions and property definitions.
    - Use parentheses to make clauses in an expression apparent
* Commenting Convention :
  + - Place the comment on a separate line, not at the end of a line of code.
    - Begin comment text with an uppercase letter.
    - End comment text with a period.
    - Insert one space between the comment delimiter (//) and the comment text.
* Language Guidelines :

<https://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>

# Software Requirement Specification

## User Requirement Specification

### Guest requirement

Guest is a person who does not have access the system. Guest can use some functions in the system. To use all functions, guest must login. These are some functions guest can use:

* Login
* Search product
* View product detail
* Make order

### Customer requirement

Customer is guest who can login to the system by his/her account and uses service of system. The customer can use some following functions:

* Logout
* Search product
* View product detail
* View personal information
* Change password
* Make order
* Edit order (before staff approve)
* View order detail
* Cancel order (before staff approve)
* Edit personal information
* Confirm order
* Get notification when:
  + Staff changes order status
  + Staff edit customer’s order status

### Staff requirement

Staff is the person who works directly with system to manage customer, product, order and bill. Staff can use some following functions:

* Logout
* Search product
* View product detail
* View personal information
* Change password
* Edit personal information
* Manage Order includes
  + Add order
  + Edit order
  + View order detail
  + Approve order
  + Change order state
* Get notification when:
  + There is a new order
  + The product quantity is under standard quantity
* Manage product includes:
  + Add new product
  + Add material to product
  + Edit product
* Manage material includes:
  + Add material
  + Edit material
  + Change material status
* Manage bill includes:
  + Add input bill
  + Edit input bill
  + Export bill
* Manage input material includes:
  + Add input material
  + Edit input material
  + Change input material status

### Manager requirement

Manager is the person who manages the system and reviews the business statistic. Manager can use some following functions:

* Logout
* Search product
* View product detail
* View personal information
* Change password
* Change product status
* Create VAT tax declaration
* Create Excise tax declaration
* Update VAT tax rate
* Update Excise tax rate
* Review income by time
* Review income by product
* Review revenue by customer
* Edit store information
* Manage staff includes:
  + Create staff account
  + Change staff account status
* Manage customer includes:
  + Change customer account status
  + Change customer account to loyal
* Manage supplier includes:
  + Add supplier
  + Edit supplier
  + Change supplier status

## System Requirement Specification

### External Interface Requirement

#### User interface

* General requirement for graphics user interface is the GUI should be simple, clear, intuitive, and reminiscent.
* The user interface uses Vietnamese language.
* The user interface uses consistent palette of colors between the text and the background.
* The user interface displays best on 1024x768-screen size.

#### Hardware Interface

* N/A

#### Software Interface

* Web application: work with Google Chrome (version 42 or above)

#### Communication Protocol

* Use HTTP protocol 1.1 for communication between the web browser and the web server.

### System Overview Use Case

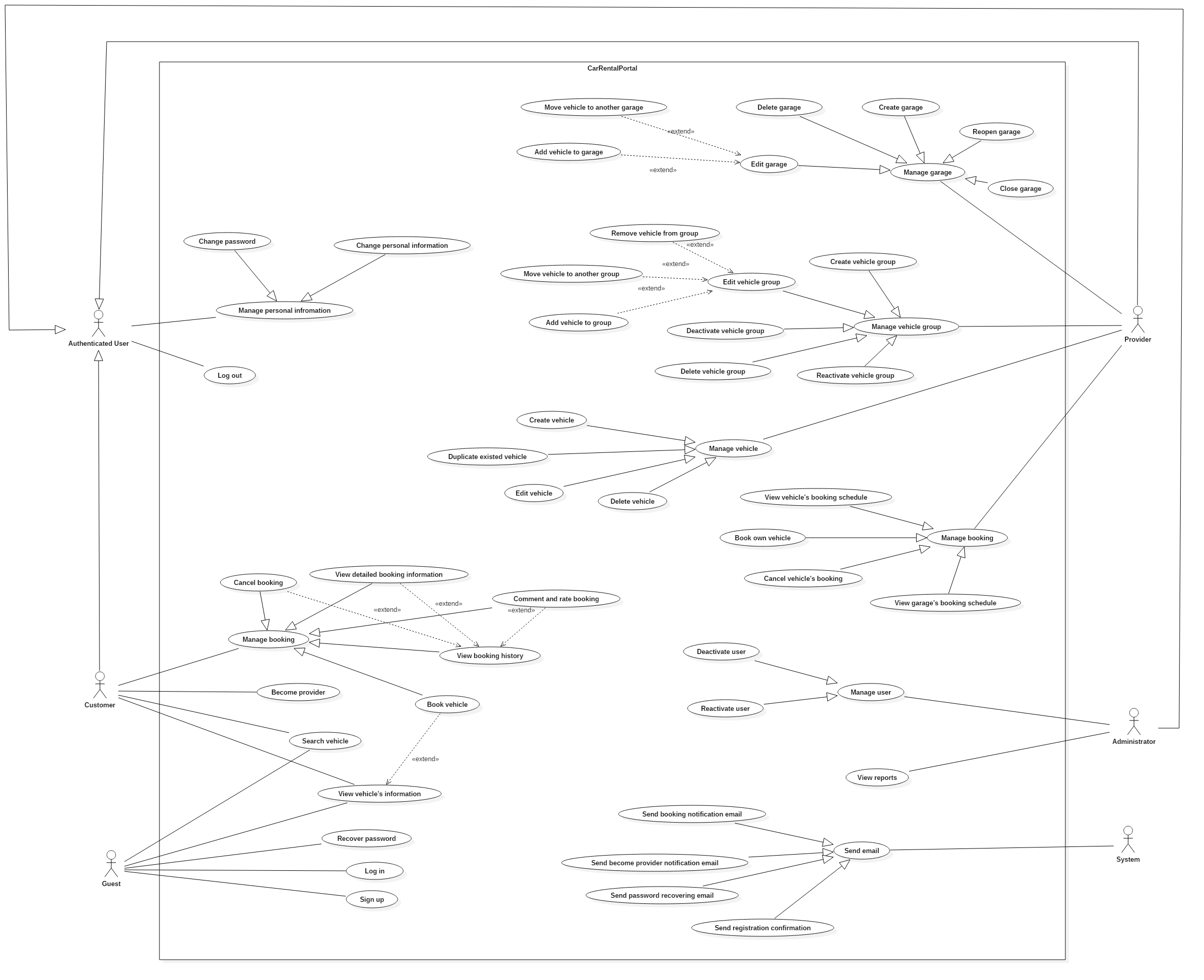


Figure 2: System Overview Use Case

### Use Cases

#### <Guest> Overview Use Case

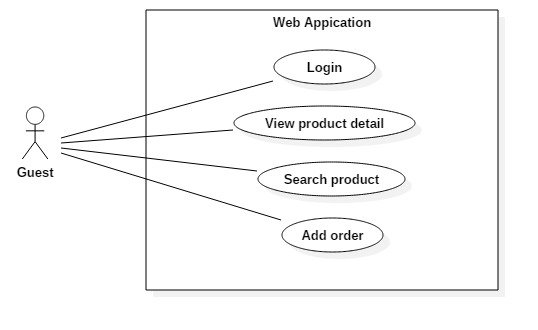


Figure 3: <Guest> Overview Use Case

#### <Authenticated user> Overview Use Case

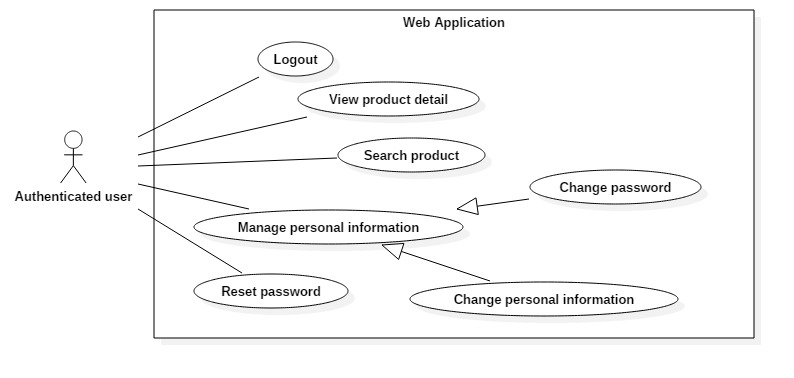


Figure 8: <Authenticated user> Overview Use Case

#### <Customer> Overview Use Case



Figure 17 <Customer> Overview Use Case

## Software System Attribute

### Usability

#### Graphic User Interface

* All the texts, labels and notifications must be written in Vietnamese.

#### Usability

* Manager, staff should need less than 2 days of training to be productive with the system.
* The system is easy to use for customer without training.

#### Installation

* The system must be easy to deploy. The bakery can deploy successfully and learn how to configure, maintain the system within two days of training.
* If there are any problems, the bakery can contact the support for help.

### Reliability

* The system must have less than 3 days downtime per one year.

### Availability

* N/A

### Security

* Privacy: Each role of user has a specific permission to interact with system.  
  System always checks the authorization and authentication before doing anything.
* Password is encrypted and save in database.

### Maintainability

* The system is divided into separated modules.

### Portability

* The system can be deployed into many type of servers those have IIS8.
* Manager, staff, customers and guests can use application on Google Chrome version 42 or above.

### Performance

* Requests from web application are responded in less than 5 seconds at 4 Mbps bandwidth speed and 1GHz Proccessing Speed of CPU.

## Conceptual Diagram

Figure 60: Conceptual Diagram

**Data Dictionary**

# Software Design Description

## Design Overview

* This document describes the technical and user interface design of BMA system. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
  + Section 2: Give an overall description of the system architecture design.
  + Section 3: Give component diagrams that describe the connection and integration of the system.
  + Section 4: Give the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
  + Section 5: Describe screen design
  + Section 6: Describe a fully attributed ERD.
  + Section 7: Describe algorithm.

## System Architecture Design

|  |
| --- |
| HTTP Request (1)    Database  Web browser  Object Relational Mapping  (ORM)  (5)  **Annotation**  Call / Send  Dependency  Query / Update  HTTP Response (6)  (4)  (3)  View Model  Entity Framework 6.0  ADO.NET Data Provider  Select (2)  URL Routing  View  Business  Model  Controller |

Figure 61: System architecture design

### Web Application architecture description

The system is developed from MVC architecture style and customized with passive model to adapt to MVC5 and Entity Framework 6.0 (EF6) technologies.

* **Model** is the part of the application that acts like a data transfer object between the system and database (POCO). Models is generated by mapping the database table (ORM) by EF6.
* **View** is the part of the application that handles the display of data. The views are created from the model data or view model data. The creation of View is under control of Controller.
* **Controller** is the part of the application that acts like event handler to handle the user interaction. Typically controller reads data from the request and calls appropriate Business’s method then selects view to generate.
* **Business** is the part of the application that do business processing to solve domain problems.
* **View Model** is the part of the application that acts like value object (VO). Typically view model contains model instances and is passed to view by controller.

## Component Diagram

Figure 62 : Component Diagram

Table 59: Component Dictionary

## Detailed Description

### Class Diagram

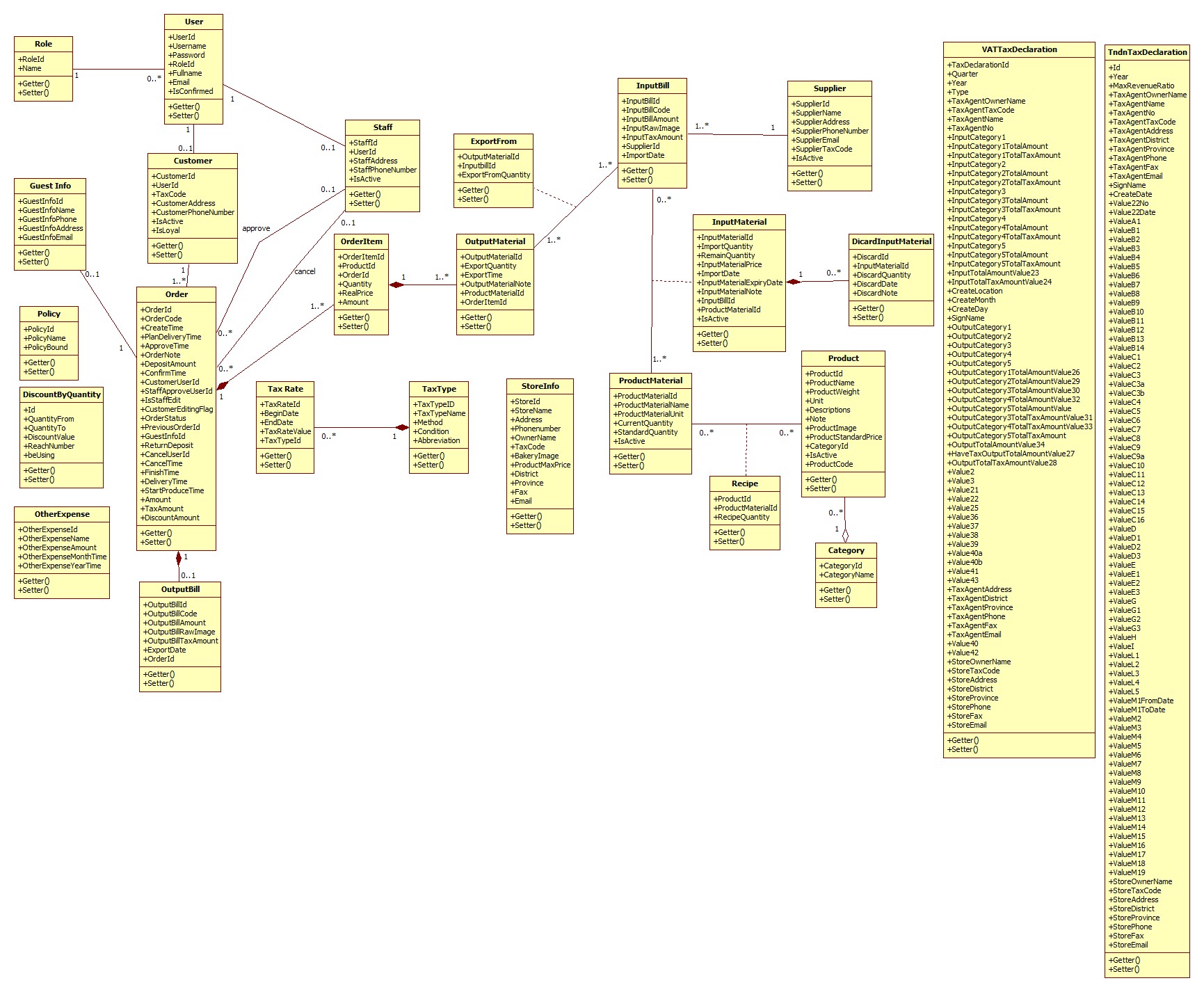


Figure 1 Class Diagram

|  |  |  |
| --- | --- | --- |
| Class dictionary: Describe Class | | |
| Class Name | **Mapping column with Conceptual diagram** | **Description** |
| Discount By Quantity | Discount By Quantity | Describe all discounts by quantity in the system |
| Other Expense | Other Expense | Describe all other expenses in the system |
| StoreInfo | StoreInfo | Describe all information of store in the system |
| Policy | Policy | Describe all policies in the system |
| Role | Role | Describe all roles in the system |
| User | User | Describe all users in the system |
| Customer | User | Describe all customers in the system |
| Staff | User | Describe all staffs in the system |
| Guest Info | Guest Info | Describe all guest information in the system |
| Orders | Orders | Describe all orders in the system |
| Output Bill | Output Bill | Describe all output bills in the system |
| Order Item | Order Item | Describe all order items of an order in the system |
| Category | Category | Describe all categories in the system |
| Product | Product | Describe all products in the system |
| Supplier | Supplier | Describe all suppliers in the system |
| Input Bill | Input Bill | Describe all input bills in the system |
| Discarded Input Material | Discarded Input Material | Describe all discard information of an input material in the system |
| Input Material | Input Material | Describe all input materials in the system |
| Export From | N/A | Describe all relationship information between output material and input bill |
| Product Material | Product Material | Describe all product materials in the system |
| Output Material | Output Material | Describe all output materials of an order item in the system |
| Recipe | N/A | Describe all of materials of each product in the system |
| Tax Rate | Tax Rate | Describe all tax rates in the system |
| Tax Type | Tax Type | Describe all tax types in the system |
| VAT Tax Declaration | VAT Tax Declaration | Describe all information in VAT tax declaration |
| Tndn Tax Declaration | Tndn Tax Declaration | Describe all information in Excise tax declaration |

Table 60 Class dictionary

### Class Diagram Explanation

### Interactive Diagram

#### Web Application

##### Guest

##### Authenticated User

##### Manager

## User Interface Design

## Database Design

### Entity relationship diagram

Figure 127 : Entity relationship diagram

### Entity Dictionary

Table 61 : Entity dictionary

Table 62 : Detail data dictionary

## Algorithms

# システムの展開とテスト

## はじめに

### 概要

本節、物理データベースの詳細とその記述は表れています。その上、BMAのテストの方法やテストプランやテストケースも記載されています。

### テストアプローチ

* ゴール: 要求を満たすため、すべてのシステムが正常に実行されていること。
* 方法: システムテスト, ブラックボックステスト。

## データベース関係ダイアグラム

### ウェブアプリケーション物理的なダイアグラム

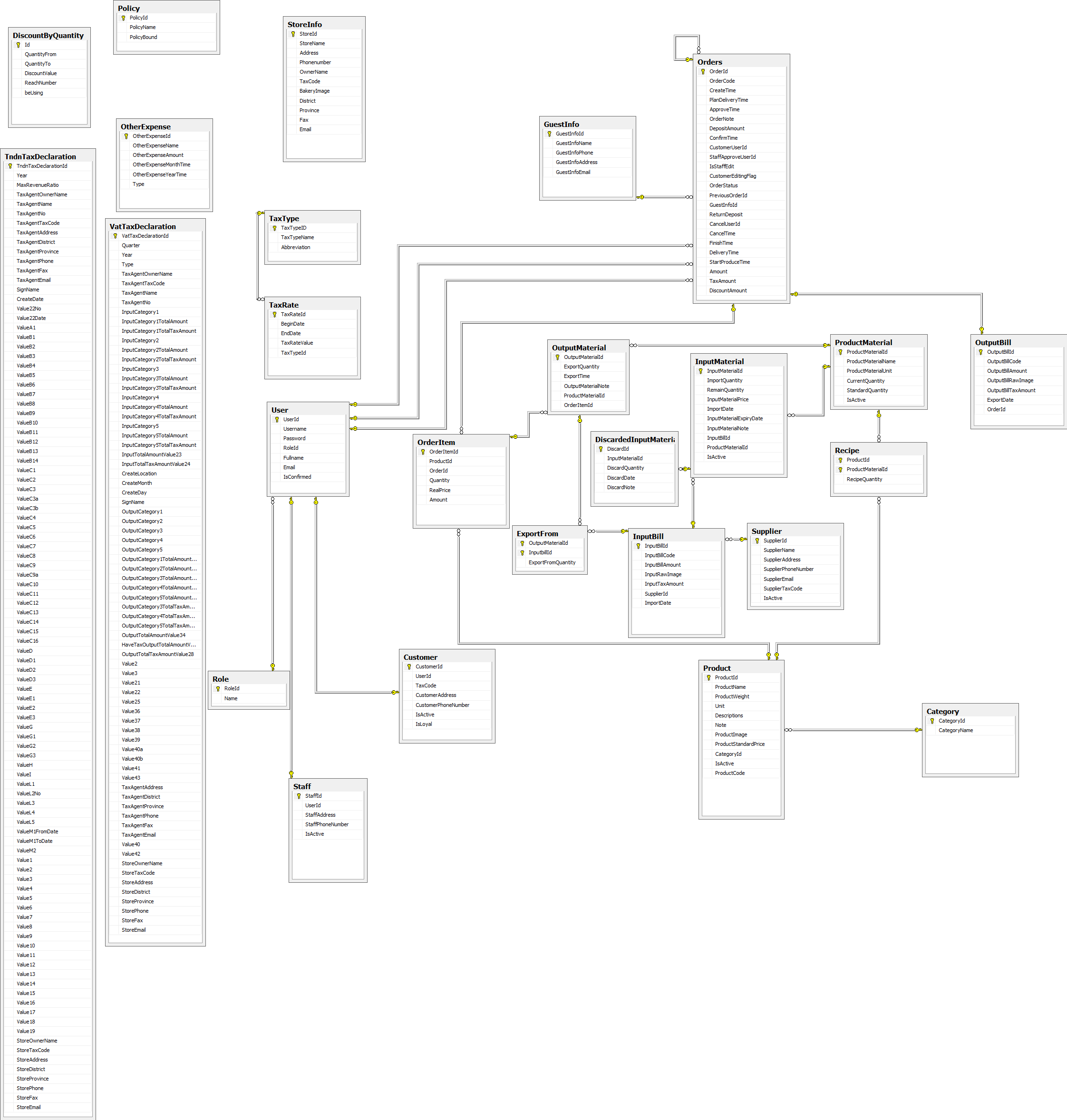


Figure 129: ウェブサイトアプリケーション物理的なダイアグラム

### 2.2 ウェブアプリケーションデータ辞書

## パフォーマンス対策

パフォーマンスを測定するために、デプロイとともにazure ホスティングでテストして、予測を行いました。

### 製品検索、メンバーのパフォーマンス

一般的に、結果はユーザが検索フォームを提出した後、5秒未満を現されます。

### 送り状リストが展示されるパフォーマンス

一般的には、ユーザーが「送り状リスト」のリンクをクリックすると、注文リストはより5秒以内に展示されます。

### 収入統計など展示されるパフォーマンス

一般的に、時間を選んで、「統計」というボタンをクリックすると、10秒以内で収入統計など展示されます。

## テスト計画

### テストする機能

システムテストはコアワークフローに基づいて行うことになります。

テストする機能:

* ユーザログイン
* ゲスト：
  + 製品を注文します
* 顧客：
  + 製品を注文します
  + 送り状を確認します
* 人員
  + 送り状を造ります
  + 送り状を承認します
  + 送り状の状態を変化します
  + 送り状を編集します
  + 入力原料を付加します
  + 入力原料を捨てます
  + 入力法案を付加します
* 管理
  + 収入統計
  + 製品別収益
  + お客様による純益統計
  + 消費税申告
  + 免許税申告

### テストされない機能

* ログ‐アウト
* プロフィールを編集
* 消費税申告
* 免許税申告
* 顧客管理
* 人員管理

## システムテストのテストケース

# ソフトウェアユーザーズマニュアル

## インストレーションガイド

### サーパ側の環境を設定

#### ハードウェア要件

**スムーズにウェブサイトを実行するてめには、以下の要件を備えたコンピュータが必要になります:**

* オペレーティングシステム: ウィンドウXPまたはウィンドウのそれ以降のバージョン。
* プロセッサ：１GHｚ以上。
* メモリ：2GB以上。
* ハードウェア：2GB以上のハードドライブスペース。
* Wifi インターネット(> 2Mbps)。

#### ソフトウェア要件

* + - ウェブサーバー ：Internet information system – IIS 8 With MVC5 enable
    - Microsoft Windows Server 2008 R2 Enterprise (Server Pack 1 64 bit): オペレイティングシ
    - .Net Framework 4.5.1
    - SQL Server 2008 R2

### サーパ側で展開

#### 展開パッケージの準備

* データベース　ファイル： BMA.mdf
* サーパーでのフォルダにソースコードファイル（BMA）をコピーする。
  + 例えば：C:\BMA

#### サーパーの構成前に展開

* **データベースの構成：**

<connectionString>

<add name = “DefaultConnection”

connectionString =”Datasource:(local);Initial Catalog=BMA;Persist Security Info = True;user id=sa;password=123456789” providerName = “System.Data.SqlClient” />

<add name = “BMAChangeDB”

connectionString =”Datasource:(local);Initial Catalog=BMA;Persist Security Info = True;user id=sa;password=123456789” providerName = “System.Data.SqlClient” />

</connectionString>

* Catalog : データベース名。
* User id :ユーザ名データベース。
* Password :パスワード

#### サーパ側でWebアプリケーションの展開

* **データベース展開：**
* ＳＱＬ Server 2008 R2 にBMA.mdfをアタックしてください。
* **ソースコードの展開：**
* オーペンIISマネージャ
* “Default Web Site”にマウスを右クリックします。“Add Application”を選びます。
  + Alias : Aliasを入力してください。例えば：ＢMA
  + Application pool : ASP.NET v4.0 を選んでください
  + Physical path:サーパーのソースコードファイルを選んでください（C:\BMA）
* “OK”ボタンをクリックしてください。
* オーペン：<http://localhost/BMA>

### クライアント側の環境の設定

#### コンピュータのための設定

* クライアントデバイスは、ウェブサイトを起動するには、以下のブラウザのいずれかを持っている必要があります：
  + Google Chrome(ダウンロードリンク):
    - <http://www.google.com/Chrome>
  + Coc Coc（コックコック）：
    - <http://www.coccoc.com/>

## ユーザーガイド

### ログイン



図面140: ログインポップアップ

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 口座を入力して |
| **2** | パスワードを入力して |
| **3** | 「Đăng nhập」ボタンをリックして |
| **4** | パスワードを忘れたら、「Quên mật khẩu」ボタンをリックして |

テーブル66: ログインポップアップ

### 製品を注文すること



図面 141:製品を注文すること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 製品を探す |
| **2** | 「Mua hàng」 ボタンをクリックして、 製品を注文する |

テーブル 67:　製品を注文する

### カートに製品を追加すること



図面 142: カートに製品を追加すること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | この製品の数量 |
| **2** | 「Thêm vào giỏ hàng 」 ボタンをくりっくして、カートに製品を追加する |

テーブル 68: カートに製品を追加する

### かーとをチェックすること



図面 143: かーとをチェックすること

|  |  |
| --- | --- |
| Step | 説明 |
| **1** | 「Giỏ hàng 」 ボタン をクリックして、製品のリストが見える |
| **2** | 「Thanh toán 」 ボタン をクリックして、カートの精細を見える |

テーブル 69: かーとをチェックすること

### カーとの精細



図面 144: カートの精細

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 製品の数量を変われる |
| **2** | 製品のリストを戻って、製品を注文して行く |
| **3** | 「Cập nhật giỏ hàng 」 ボタンをクリックして、 カートを更新する |
| **4** | 配送日をセレクトする |
| **5** | 「Đặt hàng 」ボタン をクリックして、注文を完成する |

テーブル 70: カートの精細

### 送り状を確認すること



図面 145: 送り状を確認すること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | カートの精細を戻れば、「Quay lại giỏ hàng 」 ボタン をクリックする |
| **2** | 送り状の確認のボタン |

テーブル 71: 送り状を確認すること

### 送り状に顧客の情報を充実すること



図面 146: 送り状に顧客の情報を充実すること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | これに成員をログインする |
| **2** | 成員なければ、情報を充実するところ |
| **3** | 「Quay lại giỏ hàng 」 ボタン をクリックして、送り状の確認ページを戻る |
| **4** | 口座とパスワードを入力して,「Đăng nhập 」ボタンをクリックして、送り状を完成する |
| **5** | 顧客の情報を入力して,「Hoàn tất 」 ボタンをクリックして、送り状を完成する |

テーブル 72: 送り状に顧客の情報を充実すること

### 送り状のリストを見ること



図面 147: 送り状のリストを見ること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 不足の原料のリスト |
| **2** | 新し送り状を造る |
| **3** | 送り状の精細 |
| **4** | 送り状を変わる |
| **5** | 送り状を取り消す |
| **6** | 探す |

テーブル 73: 送り状のリストを見ること

### 「人員」送り状を造ること



図面 148: 「人員」送り状を造ること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 製品 の単価 |
| **2** | 製品の数量 |
| **3** | 送り状に製品をテーキオフする |
| **4** | 送り状をキャンセルこと |
| **5** | 送り状を続く |
| **6** | 送り状に製品を追加する |

テーブル 74: 「人員」送り状を造ること

### 送り状に顧客を追加すること



図面 149: 送り状に顧客を追加すること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 顧客をセレクトする |
| **2** | 顧客なければ、口座を造る |
| **3** | お客さんの名前 |
| **4** | お客さんの口座 |
| **5** | お客さんのメール |
| **6** | お客さんの住所 |
| **7** | お客さんの電話番号 |
| **8** | お客さんの税額 |
| **9** | 入力をすべてリセット |
| **10** | 口座を造って、この口座にその送り状を追加する |
| **11** | 送り状を造るのページを戻る |
| **12** | 送り状をキャンセルする |

テーブル 75: 送り状に顧客を追加すること

### 新い顧客の口座で送り状をチェックアウト



図面 150: 新顧客の口座で送り状をチェックアウト

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 新顧客の情報 |
| **2** | 送り状の情報 |
| **3** | この送り状に原料数量 を過ごす |
| **4** | 送り状にお客さんを追加する ページを戻る |
| **5** | 送り状を造るのページを戻る |
| **6** | この送り状を確認する |
| **7** | この送り状をキャンセルする |

テーブル 76: 新顧客の口座で送り状をチェックアウト

### 入力法案 リスク



図面 151: 入力法案 リスク

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 新入力法案を造る |
| **2** | 探す |
| **3** | 入力法案 リスク |
| **4** | 入力法案 の精細を見る |
| **5** | 入力法案 の情報を変わる |

テーブル 77:入力法案 リスク

### 新入力法案を造ること



図面 152: 新入力法案を造ること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | サプライヤー をセレクトする |
| **2** | この入力法案料(VATがない) |
| **3** | VAT の料金 |
| **4** | 輸入日 |
| **5** | 入力法案の写真 |
| **6** | 入力法案 陸とを戻る |
| **7** | 新入力法案を造る |

テーブル 78: 新入力法案を造ること

### 入力原料 リスクを見る



図面 153: 入力原料 リスク

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 新入力原料を造ること |
| **2** | 探す |
| **3** | 入力原料 リスク |
| **4** | 入力原料 精細 |
| **5** | 入力原料 の情報を変わる |
| **6** | 入力原料をデリーとする |

テーブル 79: 入力原料リスク

### 入力原料 の精細を見ること



図面 154:入力原料 の精細を見ること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 入力原料の情報 |
| **2** | 入力原料をデリートする |
| **3** | 同じ入力法案の入力原料リスク |
| **4** | 入力原料の脚注 |
| **5** | 入力原料デリーとの情報 |
| **6** | 入力原料 リスクを戻る |
| **7** | 入力原料 の情報を更新する |

テーブル 80:入力原料の精細を見ること

### 新入力原料を造ること



図面 155: 新入力原料を造ること

|  |  |
| --- | --- |
| ステップ | 説明 |
| **1** | 入力原料名 |
| **2** | 入力原料の数量 |
| **3** | 入力原料の総価格 |
| **4** | 入力原料の輸入日 |
| **5** | 入力原料の有効期限 |
| **6** | 入力原料の脚注 |
| **7** | 入力原料の 入力法案 のコード |
| **8** | 入力原料 リスクを戻る |
| **9** | 入力原料を造る |

テーブル 81:入力原料を造ること

# Appendix

1. Sashimi – Modified Waterfall Model:

<http://www.waterfall-model.com/sashimi-waterfall-model>

1. UML for drawing diagram (2.0 standard)

<http://www.omg.org/spec/UML/2.0>