## 

## Software Architecture Design Document.

Version 1.0

**Son Dang Huy**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Content |
| **0.1** | 31-May | Son | Template – first draft – static diagram image |
| **0.2** | 6-Jun | Son | Mapping with architecture the driver document. |
| **0.3** | 7-Jun | Son | static diagram update and edit. Add I, II, III and IV. |
| **0.4** | 16 – Jun | Son | Complete session from I to IV. Add session V in overview. update quality attributes |
| **0.5** | 18 – Jun | Son | Add session V in detail.  Update combine view and data model view. |
| **0.6** | 20 – Jun | Son | **Complete** without C&C view. Update decomposition and layer view. |
| **0.7** | 24 - Jun | Son | Fix some error and complete. |
| **0.8** | 25 – Jun | Son | Add C&C view. |
| **0.9** | 29 – Jun | Son | Review and fix some error |
| **1.0** | 3 – July | Son | Review – release. |
|  |  |  |  |

Contents

[Software Architecture Design Document. 1](#_Toc329076804)

[I. Introduction. 5](#_Toc329076805)

[**1.1.** **Purpose.** 5](#_Toc329076806)

[**1.2.** **Reader.** 5](#_Toc329076807)

[**1.3.** **Document overview.** 5](#_Toc329076808)

[II. Project overview. 5](#_Toc329076809)

[**2.1.** **Project description.** 5](#_Toc329076810)

[**2.2.** **Project team.** 5](#_Toc329076811)

[**2.3.** **Stakeholders.** 5](#_Toc329076812)

[**2.4.** **Scope and vision.** 5](#_Toc329076813)

[III. Architecture the driver. 5](#_Toc329076814)

[**3.1.** **Key requirement.** 5](#_Toc329076815)

[**3.2.** **Business constraints.** 5](#_Toc329076816)

[**3.3.** **Technical constraints.** 6](#_Toc329076817)

[**3.4.** **Quality attributes.** 6](#_Toc329076818)

[IV. System architecture specification. 7](#_Toc329076819)

[4.1. System Context 7](#_Toc329076820)

[**4.2.** **System users.** 7](#_Toc329076821)

[4.3. Overview. 7](#_Toc329076822)

[V. Detail architect design. 7](#_Toc329076823)

[**5.1.** **Allocation view.** 8](#_Toc329076824)

[**5.1.1.** **Primary presentation.** 8](#_Toc329076825)

[**5.1.2.** **Element catalogue.** 8](#_Toc329076826)

[**5.2.** **Components and connections view.** 9](#_Toc329076827)

[**5.2.1.** **Primary presentation (Multi tier view).** 9](#_Toc329076828)

[**5.2.2.** **Element behavior.** 9](#_Toc329076829)

[**5.2.3.** **Element catalogue.** 10](#_Toc329076830)

[**5.3.** **Decomposition view.** 10](#_Toc329076831)

[**5.3.1.** **Primary presentation.** 10](#_Toc329076832)

[**5.3.2.** **Element catalogue.** 11](#_Toc329076833)

[**5.4.** **Layers view.** 12](#_Toc329076834)

[**5.4.1.** **Primary presentation.** 12](#_Toc329076835)

[**5.4.2.** **Element catalogue.** 12](#_Toc329076836)

[**5.4.3.** **Properties and relationship.** 12](#_Toc329076837)

[**5.5.** **Combine view (Decomposition and Layers view).** 12](#_Toc329076838)

[**5.5.1.** **Primary presentation.** 12](#_Toc329076839)

[**5.5.2.** **Element catalogue.** 13](#_Toc329076840)

[**5.6.** **Data model view.** 13](#_Toc329076841)

[**5.6.1.** **Primary presentation.** 13](#_Toc329076842)

[**5.7.** **Mapping.** 14](#_Toc329076843)

1. **Introduction.**
   1. **Purpose.**

Document describes architecture of system for Smart Mart, use for developing and design detail of software.

* 1. **Reader.**

|  |  |  |
| --- | --- | --- |
| No | Intended Audience | Reading Suggestions |
| 1 | Architecture & Design Team | Deployment View, C&C View, Module View |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| No | Glossary | Description |
| 1 | QA | Quality Attribute |
| 2 | OSP | Online Store Project |
| 3 | C&C | Component and Connector |

* 1. **Document overview.**

Document include 3 part, provide to reader a big picture of system architect.

* Project overview.
* Architecture the driver and architecture background.
* Architecture specification.

1. **Project overview.**
   1. **Project description.**

* **Project name: Smart Mart system.**
* **Purpose:** Develop a program can help customer manage their sale operation in store system, manage inventory, manage store’s customer and manage store’s user by using support from the computers.
* **Duration.** A month from 21st May to 25th June.
  1. **Project team.**
* 5 team members.
  1. **Stakeholders.**
  2. **Scope and vision.**

Program will help customer of store system to manage their sale operation, manage store’s customer, user and inventory

1. **Architecture the driver.**
   1. **Key requirement.**

The customer want team to develop a web application support in store management of their company. The program include 2 parts. Management and functional.

* Management, help customer manage categories in system, user account, product and category, store and customer.
* Functional provide functions support to system operation. Include, sale, make report and statistic.
  1. **Business constraints.**

|  |  |  |  |
| --- | --- | --- | --- |
| NO | Content | Constraint | Priority |
| 1 | Schedule | Develop in about 1 month. | High |
| 2 | Team members | 5 members. | High |

* 1. **Technical constraints.**

|  |  |  |  |
| --- | --- | --- | --- |
| NO | Content | Constraint | Priority |
| 1 | Platform | .NET Framework | High |
| 2 | Develop method | MVC model version 3 | High |
| 3 | Language | C# | High |
| 4 | Tool | Visual Studio 2010 SP1 Ultimate | High |
| 5 | Runtime environment | Software must install in a computer and can run on any browsers. | High |

* 1. **Quality attributes.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Attribute** | **Priority** |
| Performance | QA.01 | Update customer’s score immediately | When the score are used, the number of score used is immediately subtracted from the number of score accrued by the member. | High |
| The synchronization between the branches and center. | Moreover, in addition to the sales operation, the system is also capable of performing the statistical analysis on the sales records of all stores in near real-time manner. | High |
| Availability | QA.02 | No down time | No down time when connection with head server have been cut. | High |
| No down time | No down time when connection of POS terminal with store server have been cut or one database has been broken. Switch to use second database to ensure transactions will be continued. | High |

1. **System architecture specification.**

## System Context

Based on Team assignment document - provided information about POS-System, the implemented Smart Mart POS system will provide a numbers of function that help head manager control, monitor and report status of store system efficiently.

The following table will describes roles and responsibility of system.

* 1. **System users.**

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Manager | Manager is the person or a group have responsibility to control store system. |
| Store manager | Store manager is the person has responsibility to control local store operation. |
| Cashier | Cashier is group of account have responsibility to operate sale activities |

## Overview.



1. **Detail architect design.**
   1. **Allocation view.**
      1. **Primary presentation.**



* + 1. **Element catalogue.**

|  |  |
| --- | --- |
| Element | Description |
| Web Server | The hardware that helps to deliver Web content that can be accessed through the Internet. Running on Windows Server 2003 – support ISS version 7.  Deployed Smart Mart Web Application, control, control access flow to system. System will be deployed two web server support to backup and operate. |
| Database | Two Smart Mart databases are deployed in a hardware.  Running on Windows Server 2003, database management system (DBMS) Microsoft SQL Server 2005. Using Replication service to backup. |
| Firewall | Default firewall of web server, protect head office network and Smart Mart database. |
| Clients | Using browser to access to system, supports all popular browsers. |
| Connection to internet | Using Secure Socket Layer (SSL) to provide communication security over the Internet. |
| LAN connection | Use TCP/IP |
| Connect to database | Connect thought TCP/IP in internal network |

* 1. **Components and connections view.**
     1. **Primary presentation (Multi tier view).**



* + 1. **Element behavior.**



* + 1. **Element catalogue.**

|  |  |
| --- | --- |
| **Element** | **Description** |
| Web browser | To send request and get reply from web server. |
| Web server IIS7 | Host of Smart Mart application, a Microsoft web server base on ASP.NET framework, IIS will get requests from clients, handle it and response to clients. |
| Views | Receive request from clients, send to controller and model to handle, display GUI on brower. |
| GUI (aspx) | Display user interface on browser. (use aspx or cshtml) |
| Controller | Get data from view and model, handle it, work with data access to get data for calculating and return result to view. |
| Model | Object , store and organized data. |
| Data access | Get request from controller, access with database using LinQ to query. Return result to controller. |
| Database | Smart Mart data store. Support by Microsoft SQL server 2005 |

* 1. **Decomposition view.**
     1. **Primary presentation.**



* + 1. **Element catalogue.**

|  |  |
| --- | --- |
| Element | Description |
| Product management module | Module help program manage product in system. Create, edit and view product. |
| Categories management module | Module help program manage categories in system. Create, edit and view categories. |
| Customer management module | Module help program manage customer in system, manage customer score. View, create and edit customer |
| Store management module | Module help program manage store in system. Create, edit and view store. |
| Report management module | Module help program manage report in system. Create and view report. |
| User account management module | Module help program manage account in system. Create, edit and view account. |
| Statistic and report module | Module help program create and collect sale orders, synthesize them, make report or statistic |
| Permission control module | Module help control user in system, verify the login and logout operation and authorization. |
| Sale module | Module help cashier perform the selling operation. |
| Store functional module | Module provide functions include, add product, add price, add customer, make store report, manage store account. |

* 1. **Layers view.**
     1. **Primary presentation.**



* + 1. **Element catalogue.**

|  |  |
| --- | --- |
| Element | Description |
| Models | Contain classes associated views and controllers when there has been a change in its state, allows the views to produce updated output, and the controllers to change the available set of commands. |
| View | Contain classes help generate an output representation from model and database. |
| Controller | Contain classes to send commands to its associated view to change the view's presentation of the, it can send commands to the model to update the model's state (e.g. editing a document). |

* + 1. **Properties and relationship.**

*Presented in primary presentation.*

* 1. **Combine view (Decomposition and Layers view).**
     1. **Primary presentation.**



* + 1. **Element catalogue.**

|  |  |
| --- | --- |
| Element | Description |
| ProductModels | Model class, manage product properties |
| OrderModels | Model class, manage order of sale properties |
| AccountModels | Model class, manage account properties |
| CategoryModels | Model class, manage category properties |
| CustomerModels | Model class, manage customer properties |
| ReportModels | Model class, manage report properties |
| StoreModels | Model class, manage store properties |
| AccountController | Controller class, control login, logout and authorization. |
| SaleController | Controller class, control selling operation. |
| StoreManagementController | Controller class, control store operation, include head office operation include, manage product category, manage user, manage customer, manage store, manage report. |
| ReportController | Controller class, processing report operations, synthesis and reporting. |
| User Interface | Views class. |

* 1. **Data model view.**
     1. **Primary presentation.**



* 1. **Mapping.**