## 

## Software Architecture Design Document.

Version 1.2

**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. |
| **1.1** | 7 - July | Son | Review – fix and release. |
| **1.2** | 12 - July | Son | Mirror fix |

Contents

[Software Architecture Design Document. 1](#_Toc329552491)

[I. Introduction. 5](#_Toc329552492)

[**1.1.** **Purpose.** 5](#_Toc329552493)

[**1.2.** **Reader.** 5](#_Toc329552494)

[**1.3.** **Document overview.** 5](#_Toc329552495)

[II. Project overview. 5](#_Toc329552496)

[**2.1.** **Project description.** 5](#_Toc329552497)

[**2.2.** **Project team.** 5](#_Toc329552498)

[**2.3.** **Stakeholders.** 5](#_Toc329552499)

[**2.4.** **Scope and vision.** 5](#_Toc329552500)

[III. Architecture the driver. 5](#_Toc329552501)

[**3.1.** **Key requirement.** 5](#_Toc329552502)

[**3.2.** **Business constraints.** 5](#_Toc329552503)

[**3.3.** **Technical constraints.** 6](#_Toc329552504)

[**3.4.** **Quality attributes.** 6](#_Toc329552505)

[IV. System architecture specification. 7](#_Toc329552506)

[4.1. System Context 7](#_Toc329552507)

[**4.2.** **System users.** 7](#_Toc329552508)

[4.3. Overview. 8](#_Toc329552509)

[V. Detail architect design. 8](#_Toc329552510)

[**5.1.** **Allocation view.** 8](#_Toc329552511)

[**5.1.1.** **Primary presentation.** 8](#_Toc329552512)

[**5.1.2.** **Element catalogue.** 9](#_Toc329552513)

[**5.1.3.** **Relationship and their properties.** 10](#_Toc329552514)

[**5.2.** **Deployment style (allocation view)** 10](#_Toc329552515)

[**5.2.1.** **Primary presentation.** 10](#_Toc329552516)

[**5.2.2.** **Element catalogue.** 10](#_Toc329552517)

[**5.2.3.** **Relationship and their properties.** 11](#_Toc329552518)

[**5.3.** **Components and connections view.** 11](#_Toc329552519)

[**5.3.1.** **Primary presentation (Multi tier style).** 11](#_Toc329552520)

[**5.3.2.** **Element behavior.** 11](#_Toc329552521)

[**5.3.3.** **Element catalogue.** 12](#_Toc329552522)

[**5.3.4.** **Relationship and their properties.** 12](#_Toc329552523)

[**5.4.** **Decomposition style (module view).** 12](#_Toc329552524)

[**5.4.1.** **Primary presentation.** 12](#_Toc329552525)

[**5.4.2.** **Element catalogue.** 13](#_Toc329552526)

[**5.5.** **Layers view.** 14](#_Toc329552527)

[**5.5.1.** **Primary presentation.** 14](#_Toc329552528)

[**5.5.2.** **Element catalogue.** 14](#_Toc329552529)

[**5.5.3.** **Relationship and their properties.** 14](#_Toc329552530)

[**5.6.** **Combine view (Decomposition and Layers view).** 14](#_Toc329552531)

[**5.6.1.** **Primary presentation.** 15](#_Toc329552532)

[**5.6.2.** **Element catalogue.** 15](#_Toc329552533)

[**5.7.** **Data model view.** 15](#_Toc329552534)

[**5.7.1.** **Primary presentation.** 15](#_Toc329552535)

[**5.8.** **Mapping.** 16](#_Toc329552536)

[**5.8.1.** **Allocation view.** 16](#_Toc329552537)

[**5.8.2.** **Module view.** 17](#_Toc329552538)

[**5.8.3.** **C&C view.** 17](#_Toc329552539)

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 | Final Priority |
| 1 | Schedule | Develop in about 1 month. | High |
| 2 | Team members | 5 members. | High |

* 1. **Technical constraints.**

|  |  |  |  |
| --- | --- | --- | --- |
| NO | Content | Constraint | Final Priority |
| 1 | .NET Framework | | High |
| 2 | MVC model version 3 | | High |
| 3 | C# | | High |
| 4 | SQL database management | | High |

* 1. **Quality attributes.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Attribute** | **Priority** |
| Performance | QA.01 |  |  |  |
|  |  |  |
| Availability | QA.02 |  |  |  |
|  |  |  |

1. **System architecture overview.**

## 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.



*System context diagram.*

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



*Deployment style*

* + 1. **Element catalogue.**

|  |  |
| --- | --- |
| **Element** | **Description** |
| Server | Two servers, support windows server 2003, SQL server 2005, IIS 7, NetFramework 4.  Hardware setup  Intel Core i3 Sandy Bridge 2200 MHz, 2GB DDR3 1333, Hitachi Hard drive 640 GB SATA 3, Realtek PCIe GBE Ethernet 1 port. |
| Client machine | User ‘s browser, support any browser |
| Smart mart web application | Software be deployed on server |
| Database | Store database of smart mart system. |
| Load balancer | Use load balancing methodology to distribute workload across multiple servers to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. |
| Cluster server | Use clustering mythology to improve performance and availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability. |

* + 1. **Relationship and their properties.**

*Relationship and their properties*

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



*Multi tier style*

* + 1. **Element behavior.**



*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. **Relationship and their properties.**

*Presented in primary presentation.*

* 1. **Decomposition style (module view).**
     1. **Primary presentation.**



*Decomposition view diagram*

* + 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.**



*Layer view diagram*

* + 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. **Relationship and their properties.**

|  |  |
| --- | --- |
| Relationship | Description |
| Allow to use | A relation between 2 layers. In this case, that mean this layer can use another layer. |

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



*Combine view diagram*

* + 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.**



*Smart mart model view.*

* 1. **Mapping.**
     1. **Allocation view.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QA01P1 | Response time | Average time of each operation must less than 5 seconds. |
| Rational   * Using high performance server to handle tasks in each request more faster. | | |
| QA01P3 | System can run normally when get high number of transactions | System run normal, sequential processing each request in 1000 transactions. |
| Rational   * Using two web server at the same time to reduce the burden on system. | | |
| Availability | QA02A1 | Ensure system operate constantly | System switch to use backup database and backup web server when problem occur and Time delay when problem occur less than 30 second. |
| Rational   * Using 2 database in 2 different server, use replication service to synchronized between them. * Support 2 web server. This be the backup for the other when the problem occur. | | |
| Security | QA03S1 | Resist attack to database | Hacker use SQL injection method to access , collect information or try to break out system. Have to ensure that no any external attack can get into the database through SQL injection method. |
| Rational   * Using writing procedures method are support in SQL server 2005. * Use SSL, HTTPS protocol to communicate in system. * Use firewall to filter and only open port 80 to access though website, hacker cannot access directly to database. | | |

* + 1. **Module view.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Security | QA03S1 | Resist attack to database | Hacker use SQL injection method to access , collect information or try to break out system. Have to ensure that no any external attack can get into the database through SQL injection method. |
| Rational   * When user send request to get data from database, controller and data access classes will not use query command to get information from database but call procedure from SQL server. | | |

* + 1. **C&C view.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QA01P1 | Response time | Average time of each operation must less than 5 seconds. |
| Rational   * Using model classes, (product) data will be load from database first to prepare for sale operation, so when cashiers access system to get product information, it is will be faster than wait program load from database and display on screen. * When user load web site first time, new data will be store in cache of browser, then each time after that browser will display information faster. | | |