## 

## Software Architecture Design Document.

Version 1.5 final

**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 |
| **1.3** | 16 – July | Son | Update |
| **1.4** | 20 – July | Son | Update |
| **1.5** |  | Son | Update |

Contents

[Software Architecture Design Document. 1](#_Toc330739759)

[I. Introduction. 5](#_Toc330739760)

[**1.1.** **Purpose.** 5](#_Toc330739761)

[**1.2.** **Reader.** 5](#_Toc330739762)

[**1.3.** **Document overview.** 5](#_Toc330739763)

[II. Project overview. 5](#_Toc330739764)

[**2.1.** **Project description.** 5](#_Toc330739765)

[**2.2.** **Project team.** 5](#_Toc330739766)

[**2.3.** **Scope and vision.** 5](#_Toc330739767)

[III. Architecture the driver overview. 5](#_Toc330739768)

[**3.1.** **Key requirement.** 5](#_Toc330739769)

[**3.2.** **Business constraints.** 6](#_Toc330739770)

[**3.3.** **Technical constraints.** 6](#_Toc330739771)

[**3.4.** **Quality attributes.** 6](#_Toc330739772)

[IV. System architecture overview. 6](#_Toc330739773)

[4.1. System Context 6](#_Toc330739774)

[**4.2.** **System users.** 6](#_Toc330739775)

[4.3. Overview. 7](#_Toc330739776)

[V. Detail architect design. 7](#_Toc330739777)

[**5.1.** **Deployment view.** 7](#_Toc330739778)

[**5.1.1.** **Primary presentation.** 7](#_Toc330739779)

[**5.1.2.** **Element catalogue.** 8](#_Toc330739780)

[**5.1.3.** **Relationship and their properties.** 9](#_Toc330739781)

[**5.2.** **Components and connections view.** 9](#_Toc330739782)

[**5.2.1.** **Primary presentation (Multi tier style) – Sale feature.** 9](#_Toc330739783)

[**5.2.2.** **Element behavior.** 10](#_Toc330739784)

[**5.2.3.** **Element catalogue.** 11](#_Toc330739785)

[**5.2.4.** **Relationship and their properties.** 11](#_Toc330739786)

[**5.3.** **Decomposition view (module).** 11](#_Toc330739787)

[**5.3.1.** **Primary presentation.** 12](#_Toc330739788)

[**5.3.2.** **Element catalogue.** 12](#_Toc330739789)

[**5.4.** **Layers view.** 13](#_Toc330739790)

[**5.4.1.** **Primary presentation.** 13](#_Toc330739791)

[**5.4.2.** **Element catalogue.** 13](#_Toc330739792)

[**5.4.3.** **Relationship and their properties.** 13](#_Toc330739793)

[**5.5.** **Combine view (Decomposition and Layers view).** 14](#_Toc330739794)

[**5.5.1.** **Primary presentation.** 14](#_Toc330739795)

[**5.5.2.** **Element catalogue.** 14](#_Toc330739796)

[**5.6.** **Data model view.** 15](#_Toc330739797)

[**5.6.1.** **Primary presentation.** 15](#_Toc330739798)

[VI. Analyze. 16](#_Toc330739799)

[**6.1.** **Allocation view.** 16](#_Toc330739800)

[**6.2.** **Module view.** 17](#_Toc330739801)

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 7 parts, provide to reader a big picture of system architect.

* Introduction.
* Project overview.
* Architecture the driver overview.
* System architecture overview and background.
* Detail architect design.
* Requirement and architect analyze.
* Views mapping.

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

* **Project name: Smart Mart.**
* **Purpose:** develop system architect for customer business, system architect that satisfy customer requirements.
* **Deliverable:** 2 document, Software the driver document and software architect design for system and prototype with sale feature.
* **Duration.** A month from 21st May to 25th June.
  1. **Project team.**
* 5 team members.
  1. **Scope and vision.**

Software develop team will design architect for store system include features that described in **software the driver document**. System will help customer of store system to manage their sale operation, manage store’s customer, user and inventory

1. **Architecture the driver overview.**
   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. | Medium |
| 2 | Team members | 5 members. | Medium |

* 1. **Technical constraints.**

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

* 1. **Quality attributes.**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Quality attribute | | Final priority |
| QA1 | Performance | **QA01P1 -** The system execute operation immediately | High |
| QA2 | Availability | **QA02A1 -** System can run normally when get high number of transactions | High |
| **QA02A2 -** Ensure system operate constantly | High |
| QA3 | Security | **QA03S1 -** Resist attack to database | Medium |

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. **Deployment 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.**

|  |  |
| --- | --- |
| **Element** | **Description** |
| Internet connection | Using https protocol communicate with external network. |
| Network connection | Using TCP/IP protocol communicate in internal network. |

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



*Multi tier style*

* + 1. **Element behavior.**



*Element behavior*

* + 1. **Element catalogue.**

|  |  |
| --- | --- |
| **Element** | **Description** |
| Web browser | To send request and get reply from web server. |
| IIS 7 | Internet Information Services (IIS) - a Web server application. Use to host Smart mart ASP.NET application, handle request from clients and send to server, IIS will handle and response it to clients |
| Views | Receive request from clients, send to controller and model, display on browser. |
| Controller | Get data from view and model, handle it, work with data access to get data for calculating and return result to view. |
| Model | 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 view (module).**
     1. **Primary presentation.**



*Decomposition 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 |
| HomeController | Controller class, control home operation. |
| AccountController | Controller class, control login, logout, access filter and set permission. |
| SaleController | Controller class, control selling operation. |
| StoreController | 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. |
| SaleDataAccess | Class, have responsibility to communicate with database, get set data. |
| View | Views classes, handle data from controller and display on screen. |

* 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 |
| HomeController | Controller class, control home operation. |
| AccountController | Controller class, control login, logout and authorization. |
| SaleController | Controller class, control selling operation. |
| StoreController | 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. |
| SaleDataAccess | Class, have responsibility to communicate with database, get set data. |
| View | Views classes, handle data from controller and display on screen. |

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



*Smart mart model view.*

1. **Analyze.**
   1. **Allocation view.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QA01P1 | Response time | Average time of each operation must less than 2 seconds. |
|  | Rational   * Using load balancer to control request to 2 web server.. | | |
| Availability | QA02A1 | System can run normally when get high number of transactions | Average time of each operation of 100 transaction must less than 3 seconds. |
| Rational   * Using load balancer to load requests from user, send to web server and response back to user. | | |
| QA02A2 - QA02A3 | Ensure system operate constantly | System still running without interruption. |
| Rational   * Using two or more web server and load balancer. * Using three or more database server and clustering technology to switch database when problem occur. Sync database by using replication mirror service of SQL management. | | |
| Security | QA03S1 | Resist attack to database | Ensure that no any external attack can get into the database. |
| 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 | Ensure that no any external attack can get into the database. |
| Rational   * Account controller have responsibility to control and set permission for sign in using system. | | |