**Architecture Design Document**

**For**

**Smart Mart System**

**(SMS)**

**Prepared by K15T1-Team01**

**Van Lang University**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Content |
| 0.1 | 31-May-2012 | Son Dang | Template – first draft – static diagram image |
| 0.2 | 6-Jun-2012 | Son Dang | Mapping with architecture the driver document. |
| 0.3 | 7-Jun-2012 | Son Dang | static diagram update and edit. Add I, II, III and IV. |
| 0.4 | 16-Jun-2012 | Son Dang | Complete session from I to IV. Add session V in overview. update quality attributes |
| 0.5 | 18-Jun-2012 | Son Dang | Add session V in detail.  Update combine view and data model view. |
| 0.6 | 20-Jun-2012 | Son Dang | **Complete** without C&C view. Update decomposition and layer view. |
| 0.7 | 24-Jun-2012 | Son Dang | Fix some error and complete. |
| 0.8 | 25-Jun-2012 | Son Dang | Add C&C view. |
| 0.9 | 29-Jun-2012 | Son Dang | Review and fix some error |
| 1.0 | 3-July-2012 | Son Dang | Review – release. |
| 1.1 | 7-July-2012 | Son Dang | Review – fix and release. |
| 1.2 | 12-July-2012 | Son Dang | Mirror fix |
| 1.3 | 16-July-2012 | Son Dang | Update |
| 1.4 | 20-July-2012 | Son Dang | Update |
| 1.5 | 22-July-2012 | Son Dang | Update |
| 2.0 | 22-July-2012 | Tam Do | Make new template  Rework System Context  Rework Layered View  Reword Combine View |
| 2.1 | 26-July | Son Dang | Hotfix |
| 2.2 | 26-July | Son Dang | Update context diagram, template, C&C view and allocation view. |

Contents

[I. Introduction. 5](#_Toc331100068)

[**1.1.** **Purpose.** 5](#_Toc331100069)

[**1.3.** **Intended Audience and Reading Suggestions.** 5](#_Toc331100070)

[**1.4.** **Reference.** 5](#_Toc331100071)

[**1.5.** **Document overview.** 5](#_Toc331100072)

[1.6. Project overview. 5](#_Toc331100073)

[**1.6.1.** **Project description.** 5](#_Toc331100074)

[**1.6.2.** **Project team.** 5](#_Toc331100075)

[**1.6.3.** **Scope and vision.** 5](#_Toc331100076)

[II. Architecture the driver overview. 5](#_Toc331100077)

[**2.1.** **Key requirement.**. 5](#_Toc331100078)

[**2.2.** **Business constraints.** 5](#_Toc331100079)

[**2.3.** **Technical constraints.** 5](#_Toc331100080)

[**2.4.** **Quality attributes.** 5](#_Toc331100081)

[III. System architecture overview. 6](#_Toc331100082)

[3.1. System Context 6](#_Toc331100083)

[**3.2.** **System users.** 6](#_Toc331100084)

[3.3. Overview. 6](#_Toc331100085)

[IV. Detail architect design. 7](#_Toc331100086)

[**4.1.** **Deployment view.** 7](#_Toc331100087)

[**4.1.1.** **Primary presentation.** 7](#_Toc331100088)

[**4.1.2.** **Element catalogue.** 7](#_Toc331100089)

[**4.1.3.** **Relationship and their properties.** 8](#_Toc331100090)

[**4.1.4.** **Rationale.** 8](#_Toc331100091)

[**4.2.** **Components and connections view.** 8](#_Toc331100092)

[**4.2.1.** **Packet 1: Manage store.** 8](#_Toc331100093)

[**4.2.2.** **Packet 2: View statistic report.** 10](#_Toc331100094)

[**4.2.3.** **Packet 3: Sale transaction.** 11](#_Toc331100095)

[**4.2.4.** **Rationale.** 12](#_Toc331100096)

[**4.3.** **Decomposition view (module).** 13](#_Toc331100097)

[**4.3.1.** **Primary presentation.** 13](#_Toc331100098)

[**4.3.2.** **Element catalogue.** 13](#_Toc331100099)

[**4.4.** **Layers view.** 14](#_Toc331100100)

[**4.4.1.** **Primary presentation.** 14](#_Toc331100101)

[**4.4.2.** **Element catalogue.** 14](#_Toc331100102)

[**4.4.3.** **Relationship and their properties.** 14](#_Toc331100103)

[**4.5.** **Combine view (Decomposition and Layers view).** 15](#_Toc331100104)

[**4.5.1.** **Primary presentation.** 15](#_Toc331100105)

[**4.5.2.** **Element catalogue.** 15](#_Toc331100106)

[**4.5.3.** **Rationale.** 16](#_Toc331100107)

[**4.6.** **Data model view.** 16](#_Toc331100108)

[**4.6.1.** **Primary presentation.** 16](#_Toc331100109)

[4.6.2. Rationale. 17](#_Toc331100110)

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

This document describe the architecture design of Smart Mart System, consist of physical perspective, dynamic perspective, static perspective, use for developing and detail design.

* 1. **Scope**.

In Final Project, our team gives a description of the analysis and design of the Smart Mart System at architectural level. Then, implement and demo at least one Use Case “Sale Transaction”.

* 1. **Intended Audience and Reading Suggestions.**

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

|  |  |  |
| --- | --- | --- |
| No | Glossary | Description |
| 1 | SMS | Smart Mart System |
| 2 | ADsD | Architecture Drivers Document |
| 3 | ADD | Architecture Design Document |
| 4 | UC | Use Case |
| 5 | QA | Quality Attribute |
| 6 | BC | Business Constraint |
| 7 | TC | Technical Constraint |

* 1. **Reference.**

|  |  |  |
| --- | --- | --- |
| No. | Document | Description |
| 1 | ADsD | Describe architecture drivers for this project |

* 1. **Document overview.**

Document include 4 parts, provide to reader a big picture of system architect.

* Introduction.
* Architecture the driver overview.
* System architecture overview and background.
* Detail architect design.
  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.**.

*Please read ADsD to more detail.*

* 1. **Business constraints.**

*Please read ADsD to more detail.*

* 1. **Technical constraints.**

*Please read ADsD to more detail*

* 1. **Quality attributes.**

*Please read ADsD to more detail.*

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. |
| POS client | Use browser to access, running in the store intranet. |
| Manager client | Use browser to access system though internet. |
| 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.  Cluster server is protect by using firewall. |

* + 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. **Rationale.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QA01P2 | load capacity | Average time of calculating and show bill on browser must less than 500 milliseconds. |
|  | Rational   * Using load balancer to control request to 2 web server. | | |
| Availability | QA02A2 - QA02A3 | Ability to operate continuously | 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 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. **Components and connections view.**
     1. **Packet 1: Manage store.**
        1. **Primary presentation (Multi tier style).**



* + - 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 to database, get and set data. |
| Database | Smart Mart data store. Support by Microsoft SQL server |

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

*Presented in primary presentation.*

* + - 1. **Element behavior.**



* + 1. **Packet 2: View statistic report.**
       1. **Primary presentation (Multi tier style).**



* + - 1. **Element catalogue.**

|  |  |
| --- | --- |
| **Element** | **Description** |
| Web browser | To send request and get reply from web server. |
| Views | Binding data from controller after processing and display to browser. |
| Controller | Get request from browser, organized data with model classes and query to database through data access class. |
| Model | Store and organized data. |

* + 1. **Packet 3: Sale transaction.**
       1. **Primary presentation (Multi tier style).**



*Multi tier style*

* + - 1. **Element catalogue.**

|  |  |
| --- | --- |
| **Element** | **Description** |
| Web browser | To send request and get reply from web server. |
| Views | Binding data from controller after processing and display to browser. |
| Controller | Get request from browser, organized data with model classes and query to database through data access class. Calculate sale input values and response to browser. |
| Model | Store and organized data.   * Order model. |

* + 1. **Rationale.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Security | QA03S1 | Resist attack to database | Ensure that no any external attack can get into the database. |
| Rational   * All access to system must be authenticated after send request to controller. | | |

* 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. |
| StoreDataAccess | Data access package ,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. |
| 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. **Rationale.**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 authentication and set permission for use use system. | | |

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



*Smart mart model view.*

### Rationale.

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QA01P2 | Load capacity | Average time of calculating and show bill on browser must less than 500 milliseconds. |
| Rational   * Local product table contain name of product, default price and local price. Web application will check price of products before each working day. If that day store use local price, program will update local price in local product table. If not, local price in local product table will be zero. | | |