Smart Mart System

**Architecture Driver Document**

Version 1.6

Revision History.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Content |
| 0.1 | 28-May-2012 | Son Dang | First version |
| 0.2 | 29-May-2012 | Son Dang | Add QA scenario.  Fix spelling errors.  Update content table. |
| Tam Do | Add use-case diagram.  Add use-case description (in progress). |
| 1.0 | 5-July | Son Dang | Fix – update QA – release |
| 1.1 | 8-July | Son Dang | update QA – release |
| 1.2 | 11 - July | Son Dang | Mirror fix – AQ and priority table |
| 1.3 | 13 - July | Son Dang | Update |
| 1.4 | 18 - July | Son Dang | Update |
| 1.5 | 20 – July | Son Dang | Fix |
| 1.6 |  | Son Dang | Update |

Contents

[**I. Introduction. 4**](#_Toc330593388)

[1.1. Purpose. 4](#_Toc330593389)

[1.2. Intended audience. 4](#_Toc330593390)

[1.3. Document overview. 5](#_Toc330593391)

[1.4. Definitions, acronyms and abbreviations. 5](#_Toc330593392)

[Customer priority 5](#_Toc330593393)

[1.5. Reference. 6](#_Toc330593394)

[**II. Project overview. 6**](#_Toc330593395)

[2.1. Business object. 6](#_Toc330593396)

[2.2. Scope. 6](#_Toc330593397)

[2.3. Duration. 6](#_Toc330593398)

[2.4. Deliverable. 6](#_Toc330593399)

[**III. Functional requirement. 6**](#_Toc330593400)

[3.1. Entities description. 6](#_Toc330593401)

[**3.1.1. Manager. 6**](#_Toc330593402)

[**3.1.2. Store manager. 7**](#_Toc330593403)

[**3.1.3. Cashier. 7**](#_Toc330593404)

[**3.2.1. Use case list. 8**](#_Toc330593405)

[**3.2.2. Use case priority. 10**](#_Toc330593406)

[**3.2.3. Use case description. 11**](#_Toc330593407)

[**a. Create product. 11**](#_Toc330593408)

[**b. Edit product 11**](#_Toc330593409)

[**c. View product. 12**](#_Toc330593410)

[**d. Sale. 13**](#_Toc330593411)

[**IV. Quality attributes. 14**](#_Toc330593412)

[4.1. Qualities list and priority. 14](#_Toc330593413)

[4.2. Quality scenario. 14](#_Toc330593414)

[**4.2.1. Performance. 14**](#_Toc330593415)

[**4.2.2. Availability. 14**](#_Toc330593416)

[**4.2.3. Security. 15**](#_Toc330593417)

[**V. Constraints. 16**](#_Toc330593418)

[5.1. Business constraints. 16](#_Toc330593419)

[5.2. Technical constraints. 16](#_Toc330593420)

[5.3. Constraints priority. 16](#_Toc330593421)

# 

# Introduction.

## Purpose.

This document describe the architecture driver of Smart Mart System consist of key functional requirement, quality attributes and constraint. Support for developing architect of system.

*The development team can use this document to review the architecture of the system. The document will be also useful for future development teams*

## Intended audience.

|  |  |  |
| --- | --- | --- |
| NO | Roles | Reading suggestion |
| 1 | Requirement develop team. | Use case, constraints |
| 2 | Detail designer | Use case, constraints |
| 3 | customer | Quality attributes. |

## Document overview.

Architecture driver document include the following parts.

* Key functional requirement. Describe the key requirement of system.
* Quality attribute. Describes quality attribute and their scenario.
* Business constraints. Describes business constraints of customer operation or team develop.
* Technical constraints. Describes technical constraints of development team.

## Definitions, acronyms and abbreviations.

|  |  |  |
| --- | --- | --- |
| **No** | **Glossary** | **Description** |
| 1 | QA | Quality Attribute |
| 2 | SAD | Software architect design document |
| 3 | SD | Software architect the driver document. |

## Customer priority

|  |  |  |
| --- | --- | --- |
| Priority | Name | Description |
| **H** | **High** | The features, quality or constraint that are required by customer, must do |
| **M** | **Medium** | Develop suggested features or quality that customer approved, can be developed or not |
| **L** | **Low** | The features or quality that customer intend to develop but it are not in the project plan. |

Difficult level

|  |  |  |
| --- | --- | --- |
| Difficulty | Name | Description |
| 1 | Challenging | Really challenging, new technology, new feature or new method that develop team do it before. Have spend much time to train and apply in project. |
| 2 | Normal | Develop team know that feature or method but not always apply in work. Have to spend more time to satisfy SD and SAD than normal. |
| 3 | Easy | Develop team easy to understand and develop, satisfy SD and SAD. That is a method, feature or technology that the development team have knowledge and can apply in project. |

## Reference.

# Project overview.

## Business object.

Customer want to develop a system architect for their store system. System architect will meet the demanding requirements of customers for the quality criteria.

## Scope.

Software team develop architect for store system.

## Duration.

In one month.

## Deliverable.

* Software architect driver document, describe system architecture the driver.
* Software architect specification document, describe architecture design of system.
* Prototype.

# Functional requirement.

## Entities description.

|  |  |
| --- | --- |
| NO | Entity |
| 1 | Manager |
| 2 | Store manager |
| 3 | Cashier |

### Manager.

Manager is the person or a group have responsibility to control store system. Manager have permission to:

* Manage system user include cashier account or store manager account.
* Manage store inventory include create new product, edit product, delete product, edit general product price.
* Manage store include create new store, add product to store, remove product from store, collect and read report from store.

### Store manager.

Store manager is the person has responsibility to control local store operation.

Store manager has permission to:

* Manage local store user under Store manager control, cashier account.
* Manage sale report.
* Manage local price of product.
* Manage store’s customer include create new customer, edit customer, disable and enable customer.

### Cashier.

Cashier is group of account have responsibility to operate sale activities.

* 1. **Use case diagram and description.**



* + 1. **Use case list.**

|  |  |  |
| --- | --- | --- |
| ID | Use Case | Entities |
| UC01.01 | Create product | Manager of head office |
| UC01.02 | Edit product | Manager of head office |
| UC01.03 | View product | Manager of head office |
| UC01.04 | Remove product | Manager of head office |
| UC02.01 | Create category | Manager of head office |
| UC02.02 | Edit category | Manager of head office |
| UC02.03 | View category | Manager of head office |
| UC03.01 | Create store | Manager of head office |
| UC03.02 | Edit store | Manager of head office |
| UC03.03 | View store | Manager of head office |
| UC04.01 | Create account | Store manager |
| UC04.02 | Edit account | Store manager |
| UC04.03 | View account | Store manager |
| UC05.01 | Create customer | Store manager |
| UC05.02 | Edit customer | Store manager |
| UC05.03 | View customer | Store manager |
| UC06.01 | Sale | Cashier |
| UC07.01 | Create local price | Store manager |
| UC08.01 | Sale report | Cashier |
| UC08.02 | Store report | Store manager |
| UC08.03 | Analysis and report | Manager of head office |

* + 1. **Use case priority.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Use Case | Customer priority | Difficult level | Final priority |
| UC01.01 | Create product |  | 3 |  |
| UC01.02 | Edit product |  | 3 |  |
| UC01.03 | View product |  | 3 |  |
| UC01.04 | Remove product |  | 3 |  |
| UC02.01 | Create category |  | 3 |  |
| UC02.02 | Edit category |  | 3 |  |
| UC02.03 | View category |  | 3 |  |
| UC03.01 | Create store |  | 3 |  |
| UC03.02 | Edit store |  | 3 |  |
| UC03.03 | View store |  | 3 |  |
| UC04.01 | Create account |  | 3 |  |
| UC04.02 | Edit account |  | 3 |  |
| UC04.03 | View account |  | 3 |  |
| UC05.01 | Create customer |  | 3 |  |
| UC05.02 | Edit customer |  | 3 |  |
| UC05.03 | View customer |  | 3 |  |
| UC06.01 | Sale |  | 3 |  |
| UC07.01 | Create local price |  | 3 |  |
| UC08.01 | Sale report |  | 3 |  |
| UC08.02 | Store report |  | 2 |  |
| UC08.03 | Analysis and report |  | 2 |  |

* + 1. **Use case description.**

1. **Create product.**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC.01.01 | | |
| Use Case Name: | Create Product | | |
| Created By: | Tin | Last Updated By: | Son |
| Date Created: | May 30, 2012 | Date Last Updated: | 7-July |
| Actors: | Manager Head Office | | |
| Description: | The purpose of this use case is to allow actor to create product. | | |
| Preconditions: | The product isn’t yet existed in system. User signed in system with manager permission. | | |
| Post conditions: | Store new product record in database. | | |
| Normal Flow: | 1. Actor send create product request to server. Browser will load create screen from cache (if that is not the first time actor try to use this use case). 2. Actor input product information into system by using client browser, browser will check valid of input information. 3. Browser will send information and request to web server (data will be send to model classes, handle by controller classes and sent to database). 4. Program will check available of product in database. 5. System return result is product created success and response to actor. | | |
| Alternative Flows: | None | | |
| Exceptions: |  | | |
| Includes: | None | | |
| Priority: | None | | |
| Frequency of Use: | None | | |
| Business Rules: | 1. Product ID is made automatically. 2. Product image size must be less than 2MB, only jpeg. | | |
| Assumptions: | None | | |
| Notes and Issues: | None | | |

1. **Edit product**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC.01.02 | | |
| Use Case Name: | Edit Product | | |
| Created By: | Tin | Last Updated By: | Son |
| Date Created: | May 30, 2012 | Date Last Updated: | 7-July |
| Actors: | Manager Head Office | | |
| Description: | The purpose of this use case is to allow actor to edit product. | | |
| Preconditions: | User signed in system with manager permission. | | |
| Post conditions: | Store new update of product record in database. | | |
| Normal Flow: | 1. View and select product a product, input new or edit information of product. Browser will check valid of input information. 2. Browser will send information and request to web server (data will be send to model classes, handle by controller classes and sent to database). 3. Program will check available information in database. 4. System store new information into database and response to actor. | | |
| Alternative Flows: | None | | |
| Exceptions: |  | | |
| Includes: | None | | |
| Priority: | None | | |
| Frequency of Use: | None | | |
| Business Rules: | 1. Cannot change product ID code. 2. Product image size must be less than 2MB, only jpeg. | | |
| Assumptions: | None | | |
| Notes and Issues: | None | | |

1. **View product.**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC.01.03 | | |
| Use Case Name: | View Product | | |
| Created By: | Tin | Last Updated By: | Son |
| Date Created: | May 30, 2012 | Date Last Updated: | 7-July |
| Actors: | Manager Head Office, store manager and cashier | | |
| Description: | The purpose of this use case is to allow actor to edit product. | | |
| Preconditions: | User signed in system with any permission. | | |
| Post conditions: | Show product in detail on screen. | | |
| Normal Flow: | 1. Actor input ID code product or choose a category on category list. Choose product want to see in list. 2. Browser will send information and request to web server. (data will be send to model classes, handle by controller classes and sent to database). 3. System return result (if that is not the first time actor try to get product information, browser will load product image, category image from cache). | | |
| Alternative Flows: | None | | |
| Exceptions: |  | | |
| Includes: | None | | |
| Priority: | None | | |
| Frequency of Use: | None | | |
| Business Rules: | None | | |
| Assumptions: | None | | |
| Notes and Issues: | None | | |

1. **Sale.**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC.06.01 | | |
| Use Case Name: | View Product | | |
| Created By: | Son | Last Updated By: | Son |
| Date Created: | May 30, 2012 | Date Last Updated: | 7-July |
| Actors: | Cashier | | |
| Description: | The purpose of this use case is to allow actor to make a sale transaction. | | |
| Preconditions: | User signed in system with cashier permission. | | |
| Post conditions: | Show sale record log after finish. | | |
| Normal Flow: | 1. Select require products. Send make a sale request to server. 2. Browser will send products list and sale request to web server to handle (data will be send to model classes, handle by controller classes and sent to database). 3. Server will calculate cost and response to client. 4. Store sale record to database. | | |
| Alternative Flows: | None | | |
| Exceptions: |  | | |
| Includes: | None | | |
| Priority: | None | | |
| Frequency of Use: | None | | |
| Business Rules: | None | | |
| Assumptions: | None | | |
| Notes and Issues: | None | | |

# Quality attributes.

## Qualities list and priority.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Quality attribute | | Customer Priority | Difficulty level | Final priority | Comment |
| QA1 | Performance | **QA01P1 -** The system execute operation immediately | High | 1 | High |  |
| QA2 | Availability | **QA02A1 -** System can run normally when get high number of transactions | High | 1 | High |  |
| **QA02A2 -** Ensure system operate constantly | High | 2 | High |  |
| QA3 | Security | **QA03S1 -** Resist attack to database | Low | 2 | Medium |  |

## Quality scenario.

### Performance.

|  |  |
| --- | --- |
| **QA01P1 Performance Scenario. The system execute operation immediately** | |
| **Portion of scenario** | **Possible value** |
| Source | Store manager – Manager - cashiers |
| Stimulate | Make random 1000 operation (add product, remove product … etc) sequentially in a minute. |
| Artifact | System |
| Environment | System operating environment (runtime) – normal mode |
| Response | Show notification that “ operation execute successful” after each command. |
| Response measure | Average time of each operation must less than 2 seconds. |

### Availability.

|  |  |
| --- | --- |
| **QA02A1 Performance Scenario. System can run normally when get high number of**  **transactions** | |
| **Portion of scenario** | **Possible value** |
| Source | Cashiers |
| Stimulate | Make random 100 transactions at the same time. |
| Artifact | System |
| Environment | System operating environment (runtime) – normal mode |
| Response | Notify that transaction successfully. |
| Response measure | Average time of each operation of 100 transaction must less than 3 seconds. |

|  |  |
| --- | --- |
| **QA02A2 Availability Scenario. Ensure system operate constantly.** | |
| **Portion of scenario** | **Possible value** |
| Source | Incident at the server, lost connect to database server. |
| Stimulate | Turn off a database server. |
| Artifact | System |
| Environment | System operating environment (runtime) – normal mode |
| Response | System switch to use backup database server when problem occur immediately. |
| Response measure | System still running without interruption. |

|  |  |
| --- | --- |
| **QA02A3 Availability Scenario. Ensure system operate constantly.** | |
| **Portion of scenario** | **Possible value** |
| Source | Incident at the web server, crash web server. |
| Stimulate | Turn off a web server. |
| Artifact | System |
| Environment | System operating environment (runtime) – normal mode |
| Response | System redirect all access from crash server to another. |
| Response measure | System still running without interruption. |

### Security.

|  |  |
| --- | --- |
| **QA03S1 Usability Scenario.**  **Resist attack to database.** | |
| **Portion of scenario** | **Possible value** |
| Source | Hacker |
| Stimulate | Hacker attack to system, collect information or try to break out system. |
| Artifact | System |
| Environment | System operating environment (runtime) – normal mode |
| Response | Block the hijack command, not perform ​​the operations and notify to administrator.  Set access permission, block access not allowed. |
| Response measure | Ensure that no any external attack can get into the database. |

# Constraints.

## Business constraints.

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

## Technical constraints.

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

* 1. **Constraints priority.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Constraint | | Customer Priority | Difficulty level | Final priority | Comment |
|  | Business constraints | Develop in about 1 month. | High | 2 | Medium |  |
| 5 people in team. | High | 2 | Medium |  |
|  | Technical constraints | .NET Framework | High | 2 | Medium |  |
| MVC model version 3 | High | 1 | High |  |
| C# | High | 3 | Medium |  |
| SQL database management | High | 3 | Medium |  |