POS SYSTEM – ARCHITECTURE DRIVERS



HIT Team

Consulting

Sales

Staffing

Support

Information of document

|  |  |
| --- | --- |
| Title | Architecture Driver Document |
| Author(s) | All team |
| Reviewer(s) | All team |
| Team name | HIT Team |
| Team members | Thanh Giang, Hiep Ta, Phuc Nguyen, Giang Nguyen, Dat Tran, Huy Huynh |
| Project mentors | Ms. Huong Nguyen, Mr. Huyen Pham |
| Editor | Hiep Ta |
| Type of report | Architecture Driver Document |
| Software used | MS Word |

Document Reviewer Information

|  |  |  |
| --- | --- | --- |
| **Reviewer Name** | **Review Attendance (R/S)** | **Comments** |
|  |  |  |
|  |  |  |
|  |  |  |

Document Approver Information

|  |  |  |
| --- | --- | --- |
| **Approver Name** | **Approver Function** | **Comments** |
|  |  |  |
|  |  |  |
|  |  |  |

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date of Issue | Author(s) | Brief Description of Change |
| 1.1 | 06/03/2012 | Thanh Giang | Consume Team member’s tasks |
| 1.2 | 15/04/2012 | Giang Nguyen | Update Quality Attributes |
| 1.3 | 18/04/2012 | Thanh Giang | Use case diagram |
| 1.4 | 24/05/2012 | Hiep Ta | Use case description |
| 1.5 | 02/06/2012 | Phuc Nguyen | Update Use case description |
| 1.6 |  |  |  |
| 1.7 |  |  |  |

Table of Contents

[1. Introduction 5](#_Toc329033310)

[1.1. Purpose 5](#_Toc329033311)

[1.2. Definitions, Acronyms and Abbreviations 5](#_Toc329033312)

[1.3. Change Process 6](#_Toc329033313)

[2. Project Overview 7](#_Toc329033314)

[3. Architectural Drivers Overview 8](#_Toc329033315)

[4. Functional Requirements 9](#_Toc329033316)

[4.1. Template 9](#_Toc329033317)

[4.2. Specifications 9](#_Toc329033318)

[4.3. Use Case Modeling 10](#_Toc329033319)

[4.3.1. Domain Model 10](#_Toc329033320)

[4.3.2. Entities 10](#_Toc329033321)

[4.3.3. Use Case List 10](#_Toc329033322)

[4.3.4. Use Cases 11](#_Toc329033323)

[5. Quality Attribute Requirements 12](#_Toc329033324)

[5.1. Template 12](#_Toc329033325)

[5.2. Quality Attribute Scenarios 13](#_Toc329033326)

[6. Constraints 14](#_Toc329033327)

[6.1. Technical Constraints 14](#_Toc329033328)

[6.2. Business Constraints 14](#_Toc329033329)

[7. Prioritization 15](#_Toc329033330)

[7.1. Priority scale 15](#_Toc329033331)

[7.2. Difficulty ranking scale 15](#_Toc329033332)

[7.3. Use Cases 15](#_Toc329033333)

[7.4. Quality Attribute Scenarios 16](#_Toc329033334)

[7.5. Constraints 17](#_Toc329033335)

[7.5.1. Technical Constraints 17](#_Toc329033336)

[7.5.2. Business Constraints 17](#_Toc329033337)

[7.6. Significant Architectural Drivers 18](#_Toc329033338)

[8. Default shapes 19](#_Toc329033339)

[8.1. Dynamic perspective 20](#_Toc329033340)

[8.1.1. Element types 20](#_Toc329033341)

[8.1.2. Relationship types 20](#_Toc329033342)

[8.2. Static perspective 21](#_Toc329033343)

[8.2.1. Element types 21](#_Toc329033344)

[8.2.2. Relationship types 21](#_Toc329033345)

[8.3. Physical perspective 22](#_Toc329033346)

[8.3.1. Element types 22](#_Toc329033347)

[8.3.2. Relationship types 22](#_Toc329033348)

# Introduction

## Purpose

This document will be used to record, communicate and refine the architectural drivers for the project. This document will act as the main repository of requirements for the length of the project.

The intended audience for this document is the HIT team and the customer (here are mentor). These stakeholders are to review the document and changes will be incorporated as per the change management process of the HIT team.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| ACDM | Architecture Centric Design Method |
| Mapping | A mapping is created either between different views or between elements or relationships and groups of elements and relationships. A mapping can be hierarchical, or on the same level. Mappings between elements or relationships and groups of elements and relationships can span across views. |
| Entity | The ACDM definition for an actor which could be a person or a system. |
| Relationship Type | A relationship type is a connector in the palette, which is used to create a relationship in the canvas. |
| Relationship | A relationship is an instance of a relationship type, and used to connect element(s) on the canvas. |
| Perspective | 3 perspectives prescribed by ACDM; static, dynamic and physical. |
| View | A view is a drawing in a perspective. There can be multiple views in a perspective. |
| Design | This is the drawing the architect does in the tool. The term design is used in this document because architecture is used in many other ways which might confuse the reader. |
| Architecture | All of the diagrams including their elements and relationships as a representation of a software system. The architecture also includes the architectural drivers which are assigned to the elements and relationships. |
| Architectural drivers | These consist of functional requirements, quality attributes, business constraints and technical constraints. Can be assigned to elements, relationships and mappings. |
|  |  |

## Change Process

When making changes to the document, the date, name of person making the change and brief description of the change must be added to the *revision history* table on the second page of this document.

# Project Overview

The ACDM Architecture Tool is a tool that will allow the software architect to draw diagrams of design of the software to be developed. Detailed project goals and all relevant contextual information for this document are presented in the document.

# Architectural Drivers Overview

The architectural drivers presented in this document include:

* **Functional Requirements:** These requirements are presented in the form of specifications and use cases. These are a refinement of the requirements documented in the raw requirements specification document of step 1 ACDM.
* **Quality Attribute Requirements:** These requirements are presented in the form of quality attribute scenarios. These scenarios are based on the quality attributes documented in the raw requirements specification document of step 1 ACDM.
* **Business Constraints:** These are the business constraints documented in the raw requirements specification document of step 1 ACDM.
* **Technical Constraints:** These are the technical constraints documented in the raw requirements specification document of step 1 ACDM.

These architectural drivers will influence the architectural design and implementation of the project. Additionally, they will impact the schedule and quality of the project. As a whole these architectural drivers define the scope of the project.

# Functional Requirements

## Template



## Specifications

Prioritization scale has been defined in section 7.1.

|  |  |
| --- | --- |
| Allow designers to capture, categorize, and refine architectural drivers | **ID:** FR1 |
| **Priority:** 1 |
| **Version:** 1 |
| **Last Changed:** 11/22/2008 |
| Architecture drivers should be categorized in the following categories:   * Functional Requirement * Quality Attribute Scenarios * Technical Constraint * Business Constraint   Users cannot define their own categories.  Information to be captured for each architecture driver is listed in [lat08] (stage 2). Functional requirements should use the use case and entity templates. Quality attribute scenarios should use the 6 step template *([lat08] page 222, chapter 9, Table9.16)*. Simple templates for business and technical constraints are also present in [lat08].  The architect should be prompted to enter the architectural drivers before they create the design. This should be done in an unobtrusive way. Also, for each architecture driver, a certain list of actions should be performed (writing prose description, rationale). If these actions are not performed, the user should be notified, again in an unobtrusive way.  Refining architectural drivers will be facilitated in allowing annotations on elements. Architecture drivers will be mapped onto elements or relationships, and thus there will be a collection of annotations attached with a driver. (see requirements traceability table for related requirements)  The purpose of this feature is to have users use the tool as the primary storage for their architectural drivers. Thus the interface and features for this requirement must allow them to perform the basic tasks of adding and editing data and also take into account other features that they might need.  More detail about usage of architecture drivers in FR6. | |
| **Open Issues** | |

## Use Case Modeling

### Domain Model

|  |  |
| --- | --- |
| Domain Object | Description |
| Use case | A collection of information that the end user will enter into the tool. This collection will include: Title, ID, Description, Entities Involved, Preconditions, Primary use case flow of events, Primary use case post conditions, Alternate user case flow of events, Alternate use case post conditions. |
| Entity | A collection of information that the end user will enter into the tool. This collection will include: Name, ID, Description, Provides Assumptions, Requires Assumptions, Identified user cases. |
| Quality attribute | A collection of information that the end user will enter into the tool. This collection will include: Title of scenario, ID, Quality attribute, Description of stakeholder’s role, Source of the stimulus, Stimulus, Relevant environmental conditions, Architectural elements, System response, and Response measure. |
| Business constraint | A collection of information that the end user will enter into the tool. This collection will include: ID, Title, and Consideration. |
| Technical constraints | A collection of information that the end user will enter into the tool. This collection will include: ID, Title, and Consideration. |
| Element type | A collection of information that the end user will enter into the tool. This collection will include: Type, Name, Description, and Perspective. |
| Relationship type | A collection of information that the end user will enter into the tool. This collection will include: Type, Name, Description, and Perspective. |

### Entities

|  |  |
| --- | --- |
| **Entity name: Architect** | **Entity ID: E01** |
| **Description:**  Architects are people who use the tool to design their software intensive system they want to build. They are the only end users of the tool. Architects are assumed to be familiar with ACDM, adept at using computers, and responsible for creating and maintaining architectures based on the ACDM methodology. | |
| **Provides assumptions:**  Architects have gathered the architectural drivers and finished ACDM stage 2. | |
| **Requires assumptions:**  Architects expect no interruptions from the tool. He wants the freedom to draw the design in various ways. The only feedbacks the Architects expect from the tool are warnings if the design is not consistent and regulations from the ACDM methodology are not met. | |
| **Identified use cases:**  Architects are the single entity for this tool. Therefore, all the use cases are identified for this entity. | |

### Use Case List

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | Use Case Name | Important Level | Difficulty Level |
| User Management | | | |
| UC\_SM01 | Add New User | High | High |
| UC\_SM02 | Search/ View User List | Medium | Low |
| UC\_SM03 | View User Detail Information | Normal | Low |
| UC\_SM04 | Update User Information, Assign Authorize | High | High |
| Product Management | | | |
| UC\_PM01 | Add New Product | High | Medium |
| UC\_PM02 | Search/ View Product List | Medium | Low |
| UC\_PM03 | View Product Detail Information | Medium | Low |
| UC\_PM05 | Update Product Information | Medium | Medium |
| Bill Management | | | |
| UC\_RM01 | Add New Bill | High | High |
| UC\_RM02 | Search/ View Bill List | Medium | Low |
| UC\_RM03 | View Bill Detail Information | Medium | Low |
| UC\_RM04 | Print Bill | Medium | Medium |
| Retail Stores Management | | | |
| UC\_RSM01 | Add New Store | High | Medium |
| UC\_RSM02 | Search/ View Store List | Medium | Low |
| UC\_RSM03 | View Store Detail Information | Medium | Low |
| UC\_RSM04 | Update Store Information | Medium | Medium |
| Category Management | | | |
| UC\_CM01 | Add New Category | High | Medium |
| UC\_CM02 | Search/ View Category List | Medium | Low |
| UC\_CM03 | View Category Detail Information | Medium | Low |
| UC\_CM04 | Update Category Information | Medium | Medium |
| Member Management | | | |
| UC\_C01 | Add New Member | High | Medium |
| UC\_C02 | Search/ View Member List | Medium | Low |
| UC\_C03 | View Member Detail Information | Medium | Low |
| UC\_C04 | Update Member Information | Medium | Medium |
| UC\_C05 | View Member Point Log | High | Medium |
| Statistic | | | |
| UC\_S01 | Analysis Statistic | Low | High |
| User Computer Management | | | |
| UC\_P01 | Add New User Computer | High | Medium |
| UC\_P02 | Search/ View User Computer List | Medium | Low |
| UC\_P03 | View User Computer Detail Information | Medium | Low |
| UC\_P04 | Update User Computer Information | Medium | Medium |
| View Point | | | |
| UC\_VP | View Point | High | High |

### Use Cases

# Quality Attribute Requirements

Quality attributes are the benchmarks that describe a system’s intended behavior within the environment for which it was built. They provide the means for measuring the fitness and suitability of a product. These requirements will have an effect on the architectural design and implementation of the product.

## Template

The following template will be used to present the quality attribute scenarios for this project.

|  |  |  |
| --- | --- | --- |
| **Title of scenario:** Descriptive title if desired | **ID:** Mnemonic reference if desired | **Version:** revision number of scenario |
| **Last Changed:** when the last change was made in MM/DD/YYYY format. |
| **Quality attribute:** The one-word quality attribute characterization | **Characterization ID:** The ID of the stage 1 characterization used as a basis for this scenario | |
| **Describe stakeholder role proposing the description:** The stakeholder(s) or stakeholder communities interested in this quality attribute scenario | | |
| Source(s) of the stimulus | Description of the originating source(s) or potential sources of the stimulus or stimuli that will result in a system response. | |
| Stimulus | Phenomenon, event, situation, etc., that prompts the system or stakeholders to react in some way or take some action. It might be a user request, an event or interrupt, an error, a request for change, and so forth. | |
| Relevant environmental conditions | Description of any relevant environmental conditions that could affect the response and the measures of the response. It might include such conditions as normal operation, peak load, degraded operation, at development time, during operation, without interrupting operation, and so forth. | |
| Architectural elements | System elements affected by the stimulus. Early in the development life cycle (pre architectural design) the affected elements might not be known or may be a very high-level description. As design commences, scenarios should be refined and elements identified as they are known. | |
| System response | Desired response of the system. This should be as specific as possible because this will be how the system will be designed to respond to the stimulus. | |
| Response measure(s) | Measure by which the quality of the response will be measured. Response measures depend on the quality attribute. Response measures vary and may be in terms of person-hours, error detection, response time, recovery time, and so forth. | |
| Associated risks | In the course of analyzing the quality attribute descriptions, the architectural design team may discover risks associated with satisfying the quality attribute requirement. These will largely be technical, but may include other issues that will influence later design decisions. | |

## Quality Attribute Scenarios

The following quality attribute scenarios will be refined during the design stage, as the team gets a clearer idea of the architecture and detail design and the implications they have on these scenarios. It is most likely that new quality attribute scenarios will be added during the design phase.

|  |  |  |  |
| --- | --- | --- | --- |
| **Title of scenario:** Freedom when drawing shapes | | **ID:** QAS1 | **Version:** 1 |
| **Last Changed:** 11/22/2008 |
| **Quality attribute:** Usability | | **Characterization ID:** QA2 | |
| **Describe stakeholder role proposing the description:** End user, Customer and Maintainer | | | |
| Source(s) of the stimulus | Architect | | |
| Stimulus | Need for the tool to allow architects to concentrate on creating the architectural representation without worrying about restrictions imposed by the tool. | | |
| Relevant environmental conditions | When the architect is creating the design. | | |
| Architectural elements | The drawing canvas and shape palette subsystems of the tool. | | |
| System response | The system should not make any inputs as required when an element is added to the canvas. | | |
| Response measure(s) | The architect shouldn’t be forced to do things in a particular sequence or limited in the number or types of elements in a perspective. | | |
| Associated risks | There might be some restrictions imposed by elements that are part of a UML palette. There might be some difficulty in determining which restrictions are reasonable without restricting flexibility. | | |

# Constraints

Constraints on the system act as premade design decisions that the development team has to adhere to. Technical and business constraints for this project are provided below.

## Technical Constraints

|  |  |  |
| --- | --- | --- |
| Consideration. | ID | Technical Constraints |
| Peripheral or network hardware. | TC01 | Database server, Web server, Router |
| Commercial hardware or software products. | TC02 | Bar code readers  Keyboard  Computer |
| Tools and methods. | TC03 | Visual Studio 2010, SQL Server Management Studio 2008.  .Net Framework |
| Protocols, interfaces, standards. | TC04 | TCP/IP protocol |
| Computer operating system(s) | TC05 | Window 7 |
| Computer languages(s) | TC06 | C# ASP.Net, MVC 3.0, Entity Framework |

## Business Constraints

|  |  |  |
| --- | --- | --- |
| Consideration | ID | Business Constraints. |
| Schedule limitations. | BC01 | 1 Project Management – 120h (4h/day) |
| 2 Programmer – 120h (4h/day) |
| 1 Architect – 90h (3h/day) |
| 2 Tester – 90h (3h/day) |
| 1 Requirement -180h (6h/day) |
| Organizational restrictions and demands. | BC02 | One team with 6 members |
| Market restrictions and demands | BC03 | Produced only for ABC Company |

# Prioritization

## Priority scale

|  |  |  |
| --- | --- | --- |
| Priority (numeric) | Priority (name) | Description |
| 1 | Must Have | Must be present in the end product at all costs. |
| 2 | Nice to Have | Customer would greatly appreciate implementation of these features. |
| 3 | If There’s Time | Consider in Fall 09 semester if customer deems them important enough. |

## Difficulty ranking scale

The difficulty scale has been defined on the basis of complexity and effort. Complexity is defined as how difficult the design of a solution is and whether the team has previous experience in designing or implementing such a design. Both measures, for complexity and effort, are relative to each other.

|  |  |
| --- | --- |
| Difficulty (numeric) | Description |
| 1 | High complexity and large amount of effort required |
| 2 | High complexity or large amount of effort required |
| 3 | Moderate complexity and medium amount of effort required |

## Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case prioritizations | | | |
| ID & Title | **Stakeholder priority** | **Difficulty ranking** | **Comments** |
| UC01 - Capture functional requirement | 1 | 2 |  |
| UC02 - Capture quality attribute | 1 | 2 |  |
| UC03 - Capture technical constraint | 1 | 3 |  |
| UC04 - Capture business constraint | 1 | 3 |  |
| UC05 - Create/Remove a view | 1 | 3 |  |
| UC06 - Draw a design in a view | 1 | 1 |  |
| UC07 - Define new element or relationship type | 1 | 1 |  |
| UC08 - Edit element or relationship type | 1 | 2 |  |
| UC09 - Remove element or relationship type | 1 | 2 |  |
| UC10 - Reset toolbox | 1 | 2 |  |
| UC11 - Manage element and relationship catalogs | 1 | 3 |  |
| UC12 - Capture design rationale for view | 1 | 3 |  |
| UC13 - Capture design rationale for an element, relationship or group | 1 | 3 |  |
| UC14 - Assign/Remove an architectural driver to/from an element or relationship or a group | 1 | 1 |  |
| UC15 - View/Remove an element or relationship to which an architectural driver is assigned | 1 | 2 |  |
| UC16 - Decompose an element | 1 | 2 |  |
| UC17 - Create a mapping between different views | 1 | 2 |  |
| UC18 - Create a mapping for groups | 1 | 2 |  |
| UC19 - Delete a mapping | 1 | 3 |  |
| UC20 - Capture design rationale for a mapping | 1 | 3 |  |
| UC21 - Assign/Remove architectural drivers to/from a mapping of elements | 2 | 2 |  |
| UC22 - Use UML to draw detail design | 3 | 1 |  |
| UC23 - Warn user about unmet design obligations | 2 | 2 |  |
| UC25 - Manage tasks for the project | 2 | 2 |  |
| UC26 - Manage a project | 1 | 3 |  |
| UC27 - Export a project | 3 | 1 |  |
| UC28 - Group elements and relationships | 1 | 2 |  |
| UC29 – Ungroup elements and relationships | 1 | 2 |  |
| UC30 – View decomposition of an element | 1 | 3 |  |
| UC31 - Export an architecture project to a document format | 1 | 2 |  |

## Quality Attribute Scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Quality Attribute Scenario prioritizations | | | |
| ID & Title | **Stakeholder priority** | **Difficulty ranking** | **Comments** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Constraints

Constraints have not been prioritized in terms of importance because by definition they are of the highest priority. We have however prioritized them by difficulty, as judged by the development team.

### Technical Constraints

|  |  |  |
| --- | --- | --- |
| Technical constraint prioritizations | | |
| Description | **Difficulty ranking** | **Comments** |
| Use Java | 3 |  |
| Third party libraries need to be approved by customer | 2 | The team might not be able to use some libraries that could potentially reduce the implementation effort. |
| Must not be based on Eclipse | 2 | Using Eclipse could have reduced implementation effort. |
| Must not utilize a web browser | 3 |  |
| Coding standards will be approved by customer | 3 |  |
| Design will be reviewed and approved by customer | 3 | This will help the team in improving their design skills, and improve quality of their design. |

### Business Constraints

|  |  |  |
| --- | --- | --- |
| Business constraint prioritizations | | |
| Description | **Difficulty ranking** | **Comments** |
| Team will use ACDM during the development lifecycle | 2 | Team is unfamiliar with ACDM, thus might not properly use the methodology throughout the lifecycle. |
| Customer will be the technical advisor for the team | 3 |  |

# Default shapes

This section provides lists of element and relationship types that will be in default shapes of the tool.

## Dynamic perspective

### Element types

|  |  |  |
| --- | --- | --- |
| Name | Shape | Description |
| Process |  | Process |
| Thread |  | Thread |
| Object |  | Object |
| Data store |  | Data store |

### Relationship types

|  |  |  |
| --- | --- | --- |
| Name | Shape | Description |
| Data flow | B  A | A sends data to B |
| Event | A  B | A sends event to B |
| Interrupt | A  B | A sends interrupt to B |
| Call return | A  B | A invokes B |

## Static perspective

### Element types

|  |  |  |
| --- | --- | --- |
| Name | Shape | Description |
| Module |  | Module |
| Procedure |  | Procedure |
| Class |  | Class |
| Library |  | Library |
| Function |  | Function |

### Relationship types

|  |  |  |
| --- | --- | --- |
| Name | Shape | Description |
| Uses | B  B  A | A uses B |
| Inherits | A | A inherits from B |

## Physical perspective

### Element types

|  |  |  |
| --- | --- | --- |
| Name | Shape | Description |
| Computer |  | Computer |
| Router |  | Router |
| Sensor |  | Sensor |
| Memory |  | Memory |
| Secondary storage |  | Secondary storage |

### Relationship types

|  |  |  |
| --- | --- | --- |
| Name | Shape | Description |
| Network | A  B | A and B are connected |

**References**

[Sandcastle] - Architectural Drivers Specification

[Lat08] Lattanze, A. *Architecting Software Intensive Systems: A Practitioners Handbook*, New York, NY: Auerbach, 2008

-- The End --