Principles of Software Architectures

**Web based POST system**

Assignment 1

**Goals:**

The goal of assignment 1 is to practice

* thinking about business context and the impact it has on the architectural drivers for a software intensive system
* identifying and classifying architectural drivers
* developing quality attribute scenarios
* using this information to drive the selection of a development strategy and see how this may affect the design of a system

**Background:**

**Overview of Project**

A point-of-sale terminal (POST) is a computerized system used to record sales and handle payments; it is

typically used in a retail store. It includes hardware components such as a computer and a bar code scanner, and software to run the system (See Figure 1.1).

[](http://www.google.com.vn/imgres?q=Point-of-sale+image&um=1&hl=vi&sa=X&biw=1362&bih=583&tbs=isch:1&tbnid=VbBi9miVYIQ-VM:&imgrefurl=http://www.fitnessmarketingmuscle.com/fitness-sales-point-of-sale-systems/&imgurl=http://www.fitnessmarketingmuscle.com/wp-content/uploads/2010/03/rp5700pos.jpg&ei=Uz41TcrwDqTKcPi0kbkH&zoom=1&w=800&h=541&iact=hc&vpx=315&vpy=80&dur=523&hovh=185&hovw=273&tx=163&ty=93&oei=Uz41TcrwDqTKcPi0kbkH&esq=1&page=1&tbnh=123&tbnw=182&start=0&ndsp=19&ved=1t:429,r:1,s:0)



The purpose of this project is to create a Virtual POST system, Web based POS software can be run on any computer with an Internet connection and supported browser, without additional software.

**Goals**

This is to describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.

The goals of the POST system can be stated as

In general the goal is increased checkout automation, to support faster, better and cheaper

services and business processes. More specifically, these include:

* quick checkout for the customer,
* fast and accurate sales analysis,
* automatic inventory control.

**System description**

Your system will support two different configurations: a distributed configuration and a stand-alone configuration. Your software will support either configuration without code level changes. This means that your system must be configured at deployment to support either

configuration. In the distributed configuration, a server (one of your team’s laptops) will be used as a secure server repository and the other team members’ laptops will be the POST. Assume that the distributed configuration can support any number of POSTs.

The cashier log in POST system successfully by his/her ID and password, and record the underway (current) sale - the items purchased. The price and description of each item and current sale total will be displayed. The customer can pay by cash or credit card (this feature will be simulated for this project). Of course your system should not compromise any

Customer account information. If the customer pay by cash, the amount tendered will be input and balance due will be calculated. When the sale is committed, it will be recorded and the inventory will be reduced

The system will have an administration capability and appropriate interfaces that can be used to set and change the item type and add/remove items from the virtual POST and generally administer and configure deployed systems. The administrator can also add/remove cashiers. The administration interface will allow system administrators to easily add/remove items to/from the system and manage staffs. Your system should support local or remote administration of deployed systems.

Your system should also be esthetically pleasing as well – but not in the physical sense as the POST have in figures 1. We will not have a physical housing for the virtual POST other than the laptops themselves, but assume that you will have the laptop display (standalone or table-side) to show videos, pictures, and so forth in your product to increase the esthetic appeal of your device. Also, your device should be intuitively easy for users to use. To add a item, the system will ask the administrator to specify the type of the item, the UPC, the description, and price. If there is a problem locating a particular item, a simple, pleasing message should be displayed.

To implement your system you ***will use*** Java, SQL Server Express Edition 2005, and JDBC. Assume that the platform will be an Intel based processor running Windows. Your team should also consider using the NetBeans IDE to make software development easy (especially GUIs).

**Task 1:** Your first task should be to carefully read, identify, and discuss the architectural drivers that your team can find in the virtual POST product description. Recall that the architecture drivers include the high level functional requirements, constraints (technical and business), and quality attributes that are important to the stakeholders of the system. Identify and classify requirements you find as one of these architectural drivers (functional requirement, technical or business constraint, or quality attribute).

Initially create a rough draft of the architectural drivers. Once the team agrees to this rough

draft, then:

1. Create at least one functional use case that describes some key systemic functionality.

2. Select the top two most important quality attribute requirements (as agreed to by the team)

and describe them as a well formed quality attribute scenario using the full six part scenario

framework described in class:

1. Source of the stimulus – The entity that generated the stimulus.
2. Stimulus – A condition that affects the system.
3. Environment – The condition under which the stimulus occurred.
4. Artifact stimulated – The artifact that was stimulated by the stimulus.
5. Response – The activity that results because of the stimulus.
6. Response measure – The measure by which the system’s response will be evaluated.

**Task 2:** Identifying Scope.

Assume that your team has been assigned to design, develop, and deploy the system described in the virtual POST business plan. Given the delivery schedule described in the virtual POST product description document, identify the scope of the project. In particular:

* What functionality will you be able to deliver with the given schedule?
* What functionality will you defer – if any and why?
* Based on the overall requirements and business context, what overall development strategy will you adopt and how will it affect your design of the system.