# **HANOI UNIVERSITY Faculty of Information Technology (FIT)**

**SOFTWARE QUALITY ASSURANCE**

**System Architecture**

**Faculty**: **Information Technology**

**Module Name**: **System Analysis & Design**

**Year**: **Fall 2020**

**Topic**: **Pet Care Center Management System**

**Group: 3**

**Group Members**: **Ly Trung Kien**

**Le Duc Long**

**Ton Nu Tu Anh**

**Nghiem Thi Xuan Thuy**

**Dang Tran Trung Hieu**

**Table of Content**

1. **Introduction**
2. Purpose:

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1. Scope:

This System Architecture Document provides an architetural overviews of Amazing e-commerce Website. Amazing e-commerce Website is developed by Group 5 of SQA tutorial 2 to support online commerce.

This document has been generated directly from the Amazing e-commerce Website implemented in Analysis and Design phase. The purpose of this section extracted from Software Architecture Document template.

1. **Architectural Representation**

This document represents a series of views including use case view, logical view, process view and deployment view. There is no separate implementation view described in this document. These are views on an underlying Unified Modeling Language model developed using StarUML and draw.io.

1. **Architectural Goals and Constraints**

There are some key requirements and system constraint that have a significant bearing in the system:

* The Amazing e-commerce website must ensure complete protecction of data from both unauthorized and authorized access. All remote accesses are subject to user username and password.
* The Amazing e-commerce website will be implemented as a client-server system.
* All performance and loading requirements must be taken into consideration as the architecture is developed.
* The e-commerce website source code must comply with the coding convention, which is discovered in Code Inspection Report.

1. **Use-Case View**

This is a briefly description of use-case view of the software architecture. The use-case view is important input to the selection of the set of scenarios and use cases that are the target of an iteration. It describes the set of use cases that represents some significant functionality. It also describe the group of scenarios that have a substantial architectural coverage or that stress or illustrate a specific, delicate point of architecture.

The Amazing e-commerce Website Use case diagrams is provided in Requirement/ Amazing\_UseCaseDiagram.jpg.

The Amazing e-commerce Website top-level Use Case contain:

* Log in/Log out: Provide process for users to log in/ log out the system.
* Edit Password: Provide process for users to edit their account password
* Cart Management: Provide process for users to add product to cart/ remove product from cart.
* Edit profile: Provide process for users to edit their own profile
* Search product: Provide process for users to search product
* Order Management: Provide process for users to manage the order(Fill the order, Place order, Check the order for customer; Add order, edit order and remove order for admin)
* Product Management: Provide process for admin to manage products in the stock( Add product, Edit product information and remove product)
* Account Management: Provide process for admin to manage accounts in the database(Add account, Edit account and remove account).

1. **Process View:**

This is a briefly description of the process view of system architecture. This fields describe the task involved in the system’s execution, interactions and configurations. Also describes the allocation of objects and classes to task. Processes exist to support user registration, required transaction and information storage.

The overall process is described in amazing\_class\_diagram.png in Analysis and Design folder.

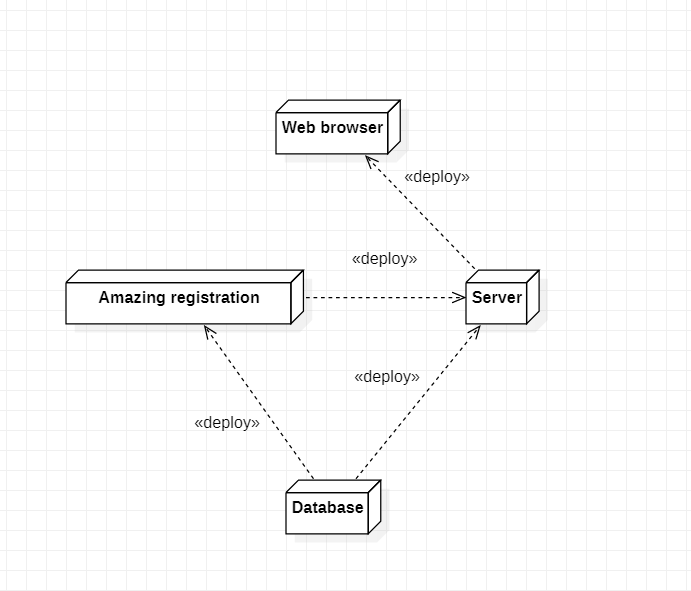
In detail, each function should be display by an activity diagram in Activity Diagrams folder and a sequence Diagram in Sequence Diagrams folder.

To be more clear, we use JavaScript Object Oriented Programming to model the object needed in functions such as Seller, Products, Customer, Payment and Cart. We also create a function to define guest and Not-registered users.

1. Seller: This object is used to store information of the one who sell product by using Amazing e-commerce software. Seller can manage their own products.
2. Products: This object is used to store information of products, which is uploaded into website in order to sell.
3. Customer: The one who access to the Amazing e-commerce Website with the intention of purchase something.
4. Payment: Save the payment after the creation of customers.
5. Cart: Store the information of chosen products for each customer.
6. Guest-or-Not-Registered-Customer: This is a function that define whether a customer is a registered customer or not.
7. **Deployment View:**

The description of the deployment view of the architecture describes the various physical nodes for the most typical platform configuration. It also describes the allocation of the tasks to physical nodes. This section is organized by physical network configuration; each such configuration is illustrated by a deployment diagram, followed by a mapping of processes to each processor.

This is the deployment diagram:



1. Database: Database of Amazing e-commerce system. Because of its size right now, it is temporarily store in our laptop, but in future, we will deploy it into cloud computing storage.
2. Amazing registration: Registration device of Amazing e-commerce system. Base on information from database, it will verify the user who want to access to the system.
3. Server: Main server of Amazing e-commerce Website.
4. Web browser: Used to display User interface of Amazing e-commerce Website
5. **Size and Performance:**

The chosen software architecture supports the key sizing and timing requirements:

* The system will support up to 200 simultaneous users against the central database at any given time at that time, and in future, this number will be up to millions.
* The system must be able to sucessfully dealing with 80% of all transactions within 2 seconds.
* The client portion shall require less than 20MB disk space and 32MB RAM. It may be become bigger in the future update with new features.
* The system shall provide access to the database within no more than 5 seconds.

The selected architecture supports the sizing and timing requirements through the implementation of a client-server architecture. The components have been designed to ensure that minimal disk and memory requirements are needed on the PC client portion.

1. **Quality:**

The software architecture supports these quality requirements:

* The desktop user-interface shall be Window 7 or more compliant.
* The user interface of Amazing e-commerce Website shall be designed for ease-of-use and be appropriate for a computer –literate user community with no addition training on the system.
* The Amazing e-commerce Website should be available 24hrs per day, 7 days per week. The maximum down-time should not be more than 5%.
* Mean Time Between Failures shall exeeds 180 hours.
* Each feature of Amazing e-commerce system should have built in online help for the user.

1. **Conclusion:**

This document produce an architectural description about the system of Amazing e-commerce Website. It specify all goals and quality requirements that Amazing e-commerce system must be satisfy. Moreover, it provide use case view, process view and deployment view for user to be more understandable about our system architecture.