E-commerce Web Application

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 1/12/2020 | 1.0 |  | Phạm Vũ Duy, Hồ Nguyễn Huy Hoàng |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Architectural Goals and Constraints 4

3. Use-Case Model 4

4. Logical View 4

4.1 Component: abc 4

5. Deployment 4

6. Implementation View 4

# Introduction

“Dang’s Company” is an E-commerce Web Application, its aim is to bring the buyers and the sellers together.

This document elaborates the software architecture document for the “Dang’s Company E-commerce Web Application”. The system architecture is abstracted into many view and components which will be explain in this document.

## 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 the significant architectural decisions which have been made on the system.

## Scope

The software architecture document applies to each static and dynamic aspect of the system.

Under the static behavior of the system, the document discusses the class diagram and other static architecture designs. Dynamic aspects of the system are elaborated using case realizations.

## Definition, Acronyms and Abbreviations

MVC – Model View Control architecture

DWA - Dang’s Company E-commerce Web Application

DB – Data Base

## Reference

<https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Definitions,%20Acronyms%20and%20Abbreviations>

<https://www.slideshare.net/PasinduTennage/sample-software-architecture-document>

<https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/artifact/ar_sadoc.htm>

## Overview

This document will present a detailed analysis of the architecture of Dang’s Company E-commerce Web Application. The further section will cover the architectural goals including the architectural constraints.

# Architectural Goals and Constraints

## Server side

The DWA will be hosted at “Heroku” JSP server. Mongo DB will be used as central database server. All communication between server and client will using HTTPS – a standard communication protocol.

## Client side

User will be able to access DWA only online. Users/Clients are expected to use a modern web browser which can support Bootstrap 4.0 to get full experience:

Recommend mobile devices browser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Google Chrome** | **Firefox** | **Safari 9.0 +** | **Android Browser 4.4+** |
| **Android** |  |  | N/A |  |
| **iOS** |  |  |  | N/A |

Desktop

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Chrome 45.0+** | **Firefox 38.0+** | **IE 10.0+** | **Microsoft Edge 12.0+** | **Opera 30.0+** | **Safari 9.0+** |
| **Mac** |  |  | N/A | N/A |  |  |
| **Windows** |  |  |  |  |  | N/A |

## Security and privacy

The central security will be handled by the member of the development team. They will be given the full access not only in the application level but also in database level. Creating account for the staff and the owner of the shop are done by administrator. When creating an account, user can choose their password and this password can be changed anytime by them. All the password are encrypted both on database or on communication process between client and server in order to ensure high level of security. The user information will only be seen by the shop owner and the administrator.

## Reliability and Availability

The system will be subjected to several testing step (Unit testing, Integration testing, System testing including Security and Performance testing) before being released in order to ensure that the system is reliable and worked as intended. The central DB can handle many numbers of request and respond to client at the same time without losing consistency and latency.

## Performance

The server responds to any request from client within the web script timeouts (30 seconds), also the system performance can depend on available hardware, network and internet connection capabilities. Therefore, the actual performance can be determined only after the system is deployed and tested.

## Portability and Reuse

The DWA is design to be a complete cosmetic website. But can be extend to sale many kinds of product. In order to maintain reusability, the web using Template Engines which can be reuse. Best practice of RUP during development combine with Mongo DB make the structured is well layered.

## Development tools

The project is the combination of many tool:

Programing tool: Visual Code

Database: MongoDB

Diagram: Draw.IO, figma

Meeting platform: Slack, Discord

Schedule: Trello

## Schedule

The development process is follows by the combination of Agile and RUP workflow. There is six sprint, each has their own workload and document:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phases** | **Interation No.** | **Tasks and Artifacts** | **Assignee** | **Start Date** | **End Date** |
| Inception | 1 | - Vision document | Nguyễn Phúc Thịnh | 1/11/2020 | 11/11/2020 |
| - Project plan | Huỳnh Nhật Nam |
| Elaboration | 1 | - Revised project plan | Huỳnh Nhật Nam | 12/11/2020 | 22/11/2020 |
| - Detailed vision document | Nguyễn Phúc Thịnh |
| - Use-case model, use-case specification | Huỳnh Nhật Nam, Nguyễn Phúc Thịnh |
| 2 | - 2nd submission use-case document | Huỳnh Nhật Nam, Nguyễn Phúc Thịnh | 23/11/2020 | 6/12/2020 |
| - Defines software architecture | Phạm Vũ Duy, Hồ Nguyễn Huy Hoàng |
| - Class diagrams | Phạm Vũ Duy, Hồ Nguyễn Huy Hoàng |
| Construction | 1 | - Revised SAD | Phạm Vũ Duy, Hồ Nguyễn Huy Hoàng | 7/12/2020 | 13/12/2020 |
| - UI prototype | Phạm Vũ Duy, Hồ Nguyễn Huy Hoàng |
| - Implementation (Source code) | All team members | 7/12/2020 | 20/12/2020 |
| - Test-plan | Mai Đăng Khánh |
| - Release (See 4.2.3 for better details) | Hồ Nguyễn Huy Hoàng | 14/12/2020 | 20/12/2020 |
| 2 | - Testing | Mai Đăng Khánh | 21/12/2020 | 27/12/2020 |

# Use-Case Model

[This section includes the use case diagrams that are already modeled and presented in the use-case specification document.]

# Logical View

[This section describes the architecture with components and relationships among them. One or several diagrams showing the architecture are provided here. For each component, describe its responsibilities and/or services that are provided for other components. Each relationship should also indicate the means of communication, such as HTTP, HTTPS, Socket, LAN, Internet, etc.

The detail of each component is provided using the subsection below.]

## Component: abc

[This section provides details for the component named “abc”. You need to include class diagrams for this component and explain key classes.

For each component, create a section like this.]

# Deployment

[Leave this section blank for PA3.]

# Implementation View

[Leave this section blank for PA3.]