System Requirements Specification:

Monitoring and Surveillance

# Introduction

The company Company.com is in the business of e-commerce in the B2C retail market. This document relates the systems supporting monitoring and surveillance of the e-commerce systems. Before release, the monitoring and surveillance will be developed in concert with the DevOps team to tune the software, system environment and delivery process.

# Purpose

The purpose of this document is to provide details of the requirements for the E-shop monitoring and surveillance. Constraints and interactions with external systems are also presented. The document is intended primarily for the software development and DevOps teams.

# Scope

# Definitions and abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| B2C | Business To Customer |
|  |  |
| REST | Representational State Transfer |
| API | Application Programming Interface |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Overall description

# Product perspective

The system is presented in figure 1 below.

A close up of a map

Description generated with very high confidence

Figure 1 – Schematic view of the system.

The user accesses the functionality via a web browser. Web servers balance the load to allow testing of new functionality on a subset of users.

The application servers provide the data for the client requests as JSON. The underlaying data is stored in relational databases.

# Product components

The E-shop is a layered system with database, API and web client.

# Business Database

Company.com uses a relational database model for storing data related to its business. The database includes most aspects of the company’s business such as Products, Suppliers, Customers and Orders.

# Business Data API

The business data is processed and exposed by a REST API. Development of the REST API is done in Java based on the Spring Boot Framework. Maven is used as project model.

# E-shop Client

Customers manage their purchases via the E-shop web client. The E-shop web client is developed as a SPA using the React JavaScript library.

# Product lifecycle

The E-shop system is to be continuously developed and updated software should be continuously delivered allowing the customers to get access to the new functionality quickly.

# Communications interfaces

The user interface (E-Shop client) communicates with the underlaying REST API using GET or POST commands.

The Business Data API server application run in a JavaEE container of an application server and conforms to the Java Servlet Specification for communication between application and server.

The Business Data API communicates with the Business Database using JDBC directly from within the application.

# Functional Requirements

# Listing of products

The customer should be able to get a listing of the all products in form of a table.

# Filtering of products

The customer should be able to filter the listing of orders.

# Put orders

The customer should be able to create a new order.

# Non-functional Requirements

# Performance Requirements

The user experience for the users of E-shop is important and should be a focus for both the Software Development and DevOps.

# Security Requirements

Best practice should be used to reduce exposure to security issues.

Routines for system patching should be established.

Exclude credentials from GitHub repositories and inject those in the build and delivery process.

Be restrictive in access to file system areas related to system or system information.

# Logging Requirements

The information from logging provide valuable information for detecting problems with the system but is also a source for continuous improvements based on usage patterns and other trends. This can be used to optimize the operations and performance and could also be a basis for future developments of the system.

Scheduled collection of log information should be setup to collect every 5 minutes under normal operations. The collected data should be appended to a file called **logarchive.csv**.

The Comma Separated Value (CSV) file should have the following columns:

**Source:** The machine generation the log e.g. Tomcat.

**Logfile name:** The name of the logfile type e.g. catalina.

**Severity:** INFO, WARN, ERROR

**Date:** Date and time.

**Message:** The text communicating what happened.

Schedule access log collection should also be setup and stored in a **access\_archive.csv**.

The Comma Separated Value (CSV) file should have the following columns:

**Source:** The machine generation the log e.g. Tomcat.

**Logfile name:** The name of the logfile type e.g. localhost\_access\_log.

**Remote host:** The remote address.

**Date:** Date and time.

**HTTP Command:** GET, POST, etc.

**Requested resource:** The requested resource e.g. /manager/html.

**Status:** The status from the response e.g. 200, 404.

# Delivery Requirements

# Software Environment

The system is to be hosted on a Linux operating system.

Within Company.com the application of choice is Tomcat 9.

HTTP servers Apache and NGINX are allowed.

Jenkins is used for CI/CD.

MySQL is the relational database used within the company.

# Continuous Deployment

The delivery should be implemented as Continuous Deployment triggered by updated master branch that is automatically deployed to the production environments.

# Analyzer Software

The Analyzer Software is not an integrated part of the E-shop but should be delivered to the system upon update of its master branch. This should be separated from the E-shop delivery.

# Miscellaneous Details

The Business Data API should run on port 8080 and be mapped to the root of the application server.

MySQL uses the default port 3306.

# Repositories

# Business Database

GitHub: <https://github.com/devops-course-2019/business-data.git>

# Business Data API

GitHub: <https://github.com/devops-course-2019/business-data-api.git>

# E-shop Client

GitHub: <https://github.com/devops-course-2019/e-shop-company-com.git>

# Analyzer Software

GitHub: <https://github.com/devops-course-2019/log-analyzer.git>

# Deliverables

# E-shop Software Components

The different software that comprise the complete system.

# System Environment

The actual environment to be used for the described E-shop applications including its delivery (CI/CD).

# System Document

Documentation of the system environment including hardware, OS, application and configuration.

# System Management Documentation

Documentation of the application management objects and how the application management should be conducted.