University of Pretoria

Bellisimo

Architectural Design Document for Assignment 1

Hlengekile Jita

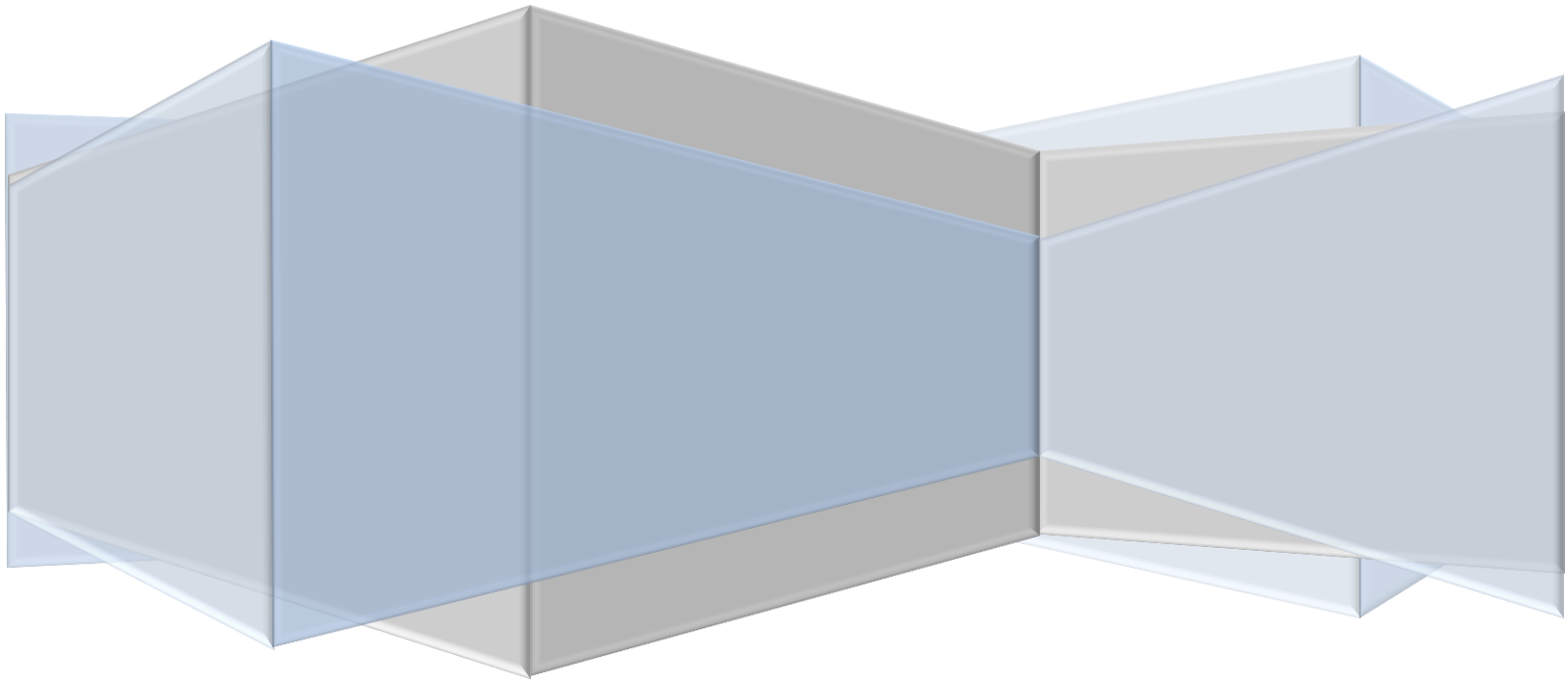


Table of Contents

Introduction

Purpose

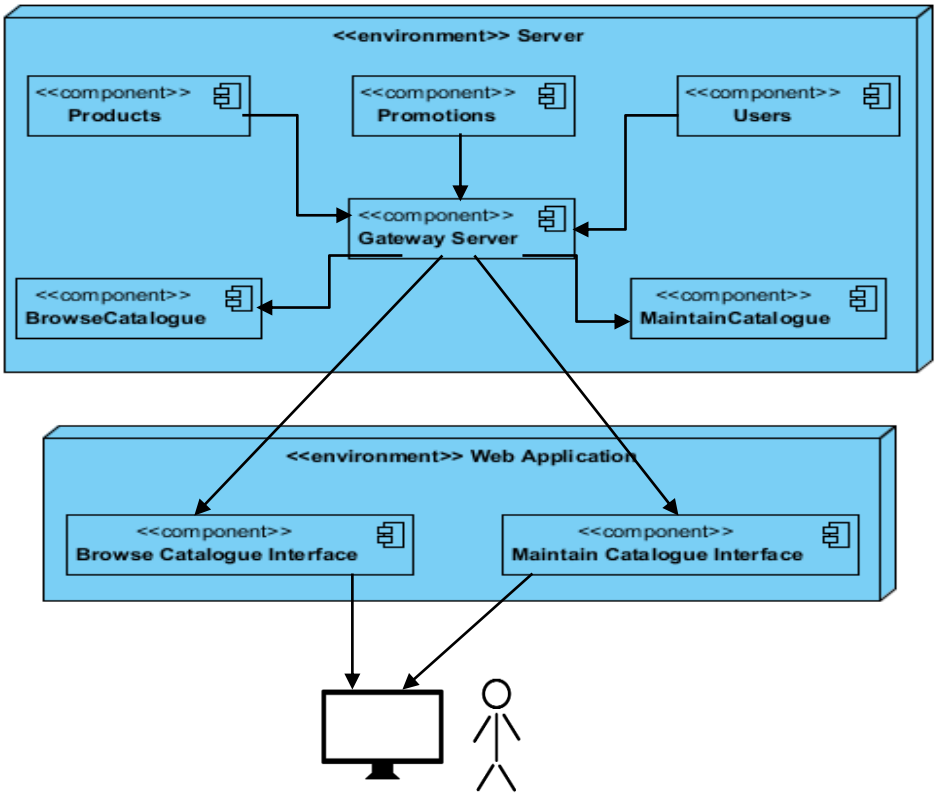
Bellisimo is an online platform that will allow customers to browse food and clothing catalogues. This document will show the architectural design of the system and explain why those architectural decisions were taken. The system is made up of a number of subsystems which will be explained in relation to the system as a whole.

Overview

This document takes the following format; firstly an overall description of the system is provided through a deployment diagram. Thereafter the diagram is explained and the various subsystems identified. Then the backend system and technologies used are explained followed by the frontend system.

Overall Description

Deployment Diagram



This system is designed using the micro-services architectural style. As seen from the above deployment diagram each subsystem or artefact is independent of the others. Instead one service, the Gateway Service is responsible for handling communication between the services. Each service or subsystem has its own data repository and provides unique functionality for the cohesive operation of the Bellisimo online catalogue.

System and Subsystems

The Bellisimo Online Catalogue is the overall system. This system will use functionality from the following subsystems:

- Products
- Promotions
- Users
- ViewCatalogue
- MaintainCatalogue

The functionality and services provided by these systems are shown in the requirements specification document. The first three subsystems will mainly involve data storage while the last two systems are the core of the system functionality and will interact with the other subsystems to provide users with the latest information about the products and their prices.

Products

This subsystem is responsible for storing the information about products. It is used by the Promotions, MaintainCatalogue and ViewCatalogue systems to display the information about the products. It uses a persistence framework and stores data in its own PostgreSQL Database.

Promotions

Promotions subsystem is responsible for tagging items/products that are on sale. The data stored here is linked with the products subsystem by querying for product IDs so that when a user views information about a product, if there is a promotion, the information about the promotion which matches the product ID such as its, promotion price, start and end dates is fetched from this subsystem. This system also uses a persistence framework.

Users

This system is mainly responsible for authenticating administrators and managing user sessions. The administrators will be predefined and stored in a database. The system will simply verify the user is an administrator by checking if they are in the database and have the correct password. For the guest users they will be given a session variable and temporarily registered in the user's database while they browse the catalogue.

MaintainCatalogue

This subsystem provides CRUD functionality to the administrator to make changes to the Products and Promotions Databases. It is an interactive system type as it allows the user to interact with the backend and receive and edit information with the backend system.

ViewCatalogue

This subsystem interacts with the Products and Promotions subsystems and fetches information about the products and their specials and displays them in a catalogue. In

addition this service provides searching and filtering functionality. This follows the client-server design as it just receives information from the other services.

Backend System

For the backend of the Bellisimo system we use the Spring framework which allows you to create services using Spring Boot then integrate them using Spring cloud and Netflix's OSS Zuul. The support services that will be created using Zuul and Spring cloud is a discovery server and edge server that will help the services interact.

Frontend Application

On the frontend, we have an Angular2 application that uses Html and Bootstrap for the design of the web application. This system will only have a web application and no mobile application.

The web app will have two interfaces, one for a normal user to browse the catalogues and one for an administrative user who will be able to make changes to the catalogues. The Angular app will interact with the Gateway Application provided in our microservices spring boot backend. The application will merely display information from the back-end and does not provide any unique functionality of its own.