University of Pretoria

Bellisimo

Software Requirements Specification Document for Assignment 1

Hlengekile Jita

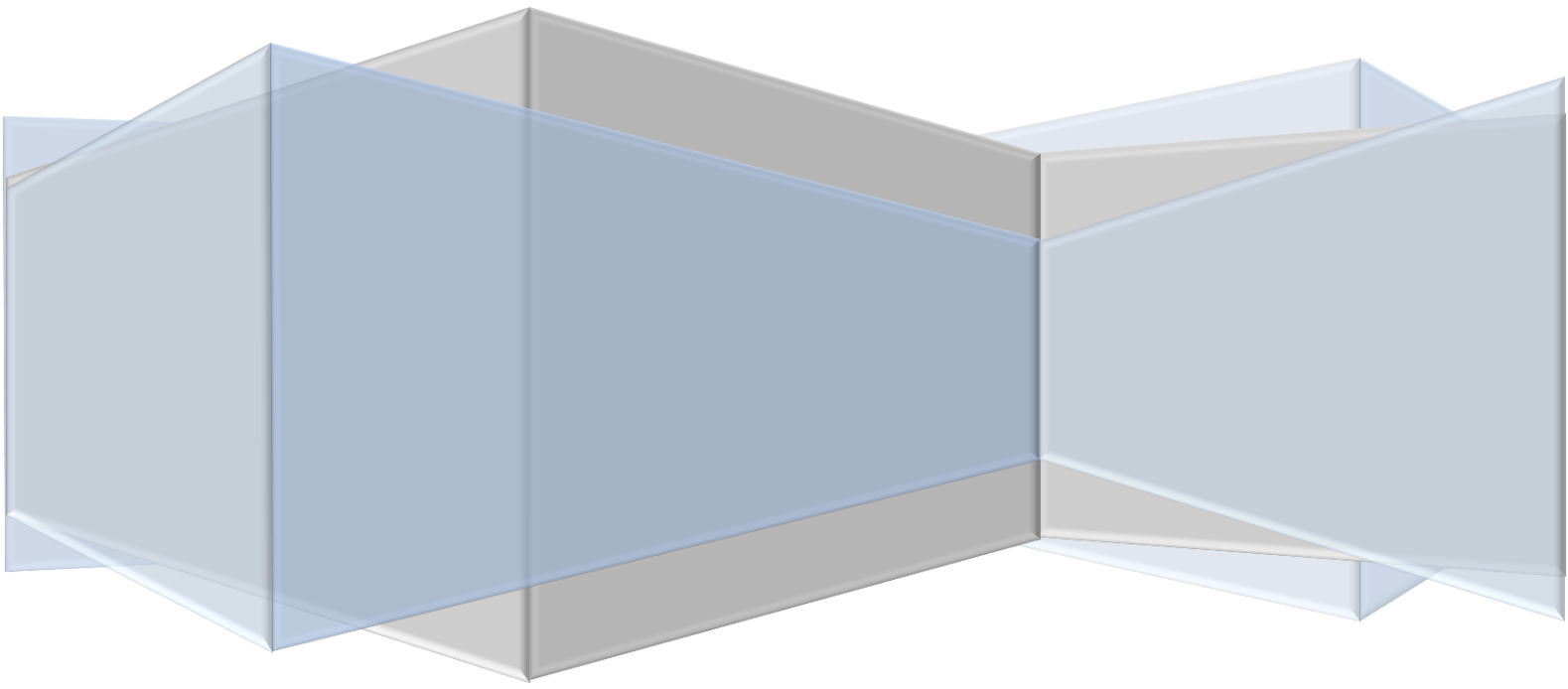


Table of Contents

Introduction

Purpose

The purpose of this document is to describe the Bellisimo online catalogue system. The system is online platform that will allow customers to browse food and clothing catalogues. This document will describe the scope and detail the functional and non-functional requirements of the system.

Scope

This Software Requirements Specification document details the requirements of the Bellisimo system. Bellismo is an online product catalogue that allows users to browse food and clothing products with their prices. In addition to that information about specials and promotions will be provided on the system. Users who are viewing the catalogue will be able to interact with the catalogue by being allowed to select items, search and filter through the catalogue. Administrative users will be able to modify the catalogue by adding, updating and deleting information about the products and the specials. The system will have to stay up-to-date and always provide the latest information about the products and their prices.

Overview

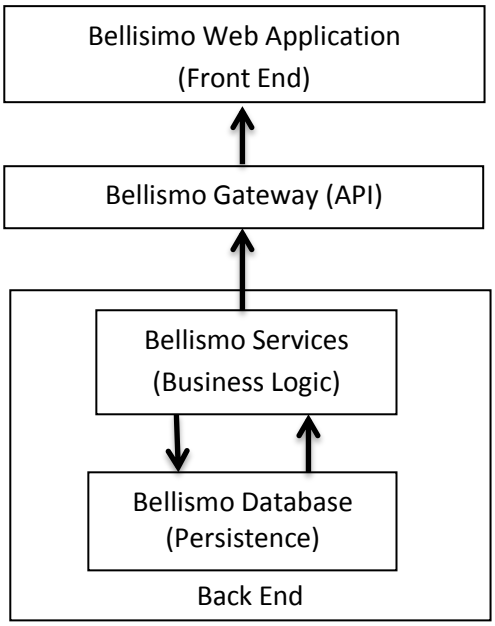
The rest of this specification is structured as follows; firstly an overall description of the product is given. In this section a product perspective is given by describing the interfaces of our system as well as the memory requirements and the operations of the system. We also describe the functions and user characteristics of Bellisimo. This section is concluded with the assumptions and dependencies that are implied by this specification. Thereafter we give the specific requirements of the system where we describe the functional requirements by giving the system use-cases and modelling the interaction between the users and the system we then also give the system attributes by describing the non-functional requirements.

Overall Description

Product Perspective

System Interface

The following system diagram shows the overall system and how each of the modules interact with each other. The front end or web application will be the users interface into the system while the gateway provides an interface for the client application to access the systems services. On the back end, this is where the core functionality of the system will lie, here the business functionality is provided and a database is used for the storage of data needed by these functions.



User Interface

The web application will need to have two separate interfaces, one for normal users who are just browsing the catalogues and another one for administrative users who will be able to maintain and make changes to the catalogues.

The catalogue browsing interface will display a list of all the products with an image and the product details such as its price and description. This screen will also have a search bar and filtering options in a side bar. Based on the search text or filtering options the user provides the list of products displayed will be changed to display only those that meet the criteria.

The catalogue maintenance interface will start with a login screen to allow the administrator to login or register as a new administrator. It will also display a list of all the products and their information but there will also be buttons with the products for edit and delete. Above this list of products there will be buttons for Add Special and Add Promotion. These buttons together with the product edit button will take the user to different screens which will allow

them to input information for the new product or promotion or modify existing information about the product they are editing.

Software Interface

The software interface is the API that will allow application(s) to communicate with the Bellisimo Software services. This interface will provide access to the application while encapsulating the actual business logic. This will allow the necessary interaction between the system and the user and the system and other systems. There will be three main software interfaces on this system, the client application, the system services and the database. These systems will interface with each other; the client application with the system through a routing gateway, and the system with the database through a persistence API. This interaction is shown in the systems interface section above.

Communications Interface

The communication within the system will take place using the REST pattern and the HTTP protocol. By designing RESTful services, the services will be able to interact with each other in a lightweight scalable manner. This is because the services are given a uniform interface that can be used by any other service. The services will also be able to send and receive messages by sending HTTP requests and receiving HTTP responses. All the system services will be designed in this way and will communicate with the HTTP requests and responses, that is, the system services, the gateway application and the client application will communicate in this way.

Memory

The information used in Bellisimo will be stored in a database on the backend. The information that needs to be stored is the information about the products and specials and the details of the administrative users. On the frontend there will be no need for memory as the application will query the information from the backend when it requires it.

Operations

The operations that will be provided by the system are as follows:

- View the catalogue of products
- Search the catalogue of products
- Filter the catalogue of products
- Add a product to the catalogue
- Remove a product from the catalogue
- Modify a product in the catalogue
- Add a promotion to a product in the catalogue
- Add a promotion to a group of promotions in the catalogue

Product Functions

The main function of Bellismo will be to provide people with an interactive catalogue of products in which they can see the products image and price.

- Browse Catalogue
  This function will allow users to scroll through the list of products, where for each product there will be an image and a price. This function will also allow a user to search for a specific product or specify the types of products they want to see by filtering.

- Maintain Catalogue
  This function will allow users to modify the catalogue by adding, removing and editing products. This function will also allow a user to add a promotion or special.

User Characteristics

The users of this system will be people who are either browsing for products and specials or modifying the products or specials. These two types of users are shown in the system through the following user roles

User roles

- Shopper
  A shopper is a normal user who is just browsing through the catalogue. This user will be anonymous so no information on them will be stored. This user will access the system through the browse catalogue user interface.
- Administrator
  An administrator is a user who is able to make changes to the catalogue. For this user, we will store the information necessary to authenticate them such as their email address and a password. This user will access the system through the main catalogue user interface.

Assumptions and dependencies

The design of this system assumes and depends on the following: Administrative users are people who either own a shop or own the catalogue; they have administrative access to the system for the right reason. This implies that we will not perform any checks before registering an administrative user to verify they are actually allowed to change the catalogue, simply because that is out of the scope of this system. It is also assumed the users will be using a web enabled device to access the application as the system is dependent on an Internet connection.

Specific Requirements

Functional Requirements

Use Cases

- UC1: A user can view the catalogue
- UC2: A user can filter the catalogue
- UC3: A user can search the catalogue
- UC4: A user can login as an administrator
- UC5: An administrative user can add a product to the catalogue
- UC6: An administrative user can remove a product from the catalogue
- UC7: An administrative user can update a product in the catalogue
- UC8: An administrative user can add a promotion to the catalogue

Actor-System Interaction Models

UC1: A user can view the catalogue

Precondition: User access the Bellisimo Web App through a URL	
Actor: Shopper	**System**
	0: The system shows the list of all the products and their prices
1: The user sees the information about the products	

UC2: A user can filter the catalogue

Precondition: Shopper is viewing the catalogue.	
Actor: Shopper	**System**
	0: Filtering options are displayed such as by price or promotions
1: User selects their filtering options	2: The system retrieves the information about only the products that meet the filtering criteria
3: The user sees the information about the products that meet the filtering criteria	

UC3: A user can search the catalogue

Precondition: Shopper is viewing the catalogue	
Actor: Shopper	**System**
	0: Search text box is displayed
1: User types in their search text	2: System retrieves information about the products that contain matching text in the product name
3: User sees which products match their search criteria	

UC4: A user can login as an administrator

Precondition: User accesses the Bellisimo Web Administrative App through its URL	
Actor: User	**System**
	0: System displays a login screen with username and password input fields
1: User enters their login details	2: System checks if a user with that username exists
	3: System tells the user that their username is incorrect
4: User sees that their username is incorrect	3: System checks if the passwords match
	4: System tells user that their password is incorrect
5: User sees that their password is incorrect	4: System logs in user and displays maintenance interface
5: User gains access to the system and sees the maintenance interface	

UC5: An administrative user can add a product to the catalogue

Precondition: User selects add product button	
Actor: Administrative User	**System**
	0: Add product screen is displayed where product information can be entered
1: User enters information about the new product	2: System creates a new product
	3: New product is persisted to the database
	4: System tells user product added

UC6: An administrative user can remove a product from the catalogue

Precondition: User selects remove button on a product	
Actor: Administrative user	**System**
	0: System asks user if they are sure they want to remove the product
1: User confirms they want to delete the product	2: System removes product from the database
	3: System tells user product removed

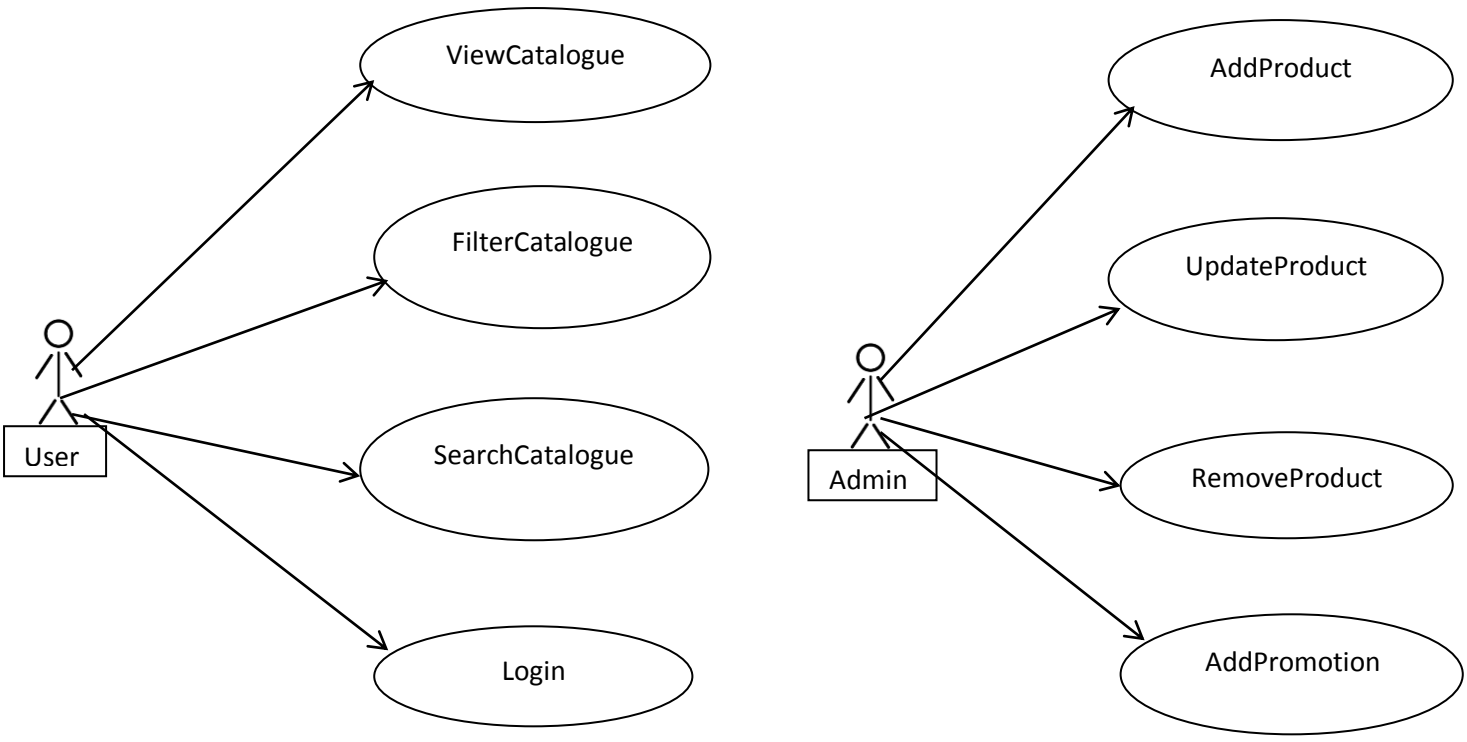
UC7: An administrative user can update a product in the catalogue

Precondition: User selects edit button on a product	
Actor: Administrative user	**System**
	0: Modify product screen is displayed where product information is displayed in edit fields
1: User makes changes to the relevant fields	2: System updates the changed fields in the database
	3: System tells user product updated

UC8: An administrative user can add a promotion to the catalogue

Precondition: User selects add promotion button	
Actor: Administrative User	**System**
	0: System asks the user if they want to add a promotion to an individual product or group of products
1: User selects individual product	2: System tells user to select the product they want to add the promotion to
3: User selects a product from the catalogue	
1: User selects group of product	2: System tells user to the group they want to add the product to
3: User selects a product group from list of groups	
	4: System shows add promotion screen with input fields
5: User enters the promotion start and end dates as well as the promotion price	6: System creates promotion
	7: Promotion is persisted to the database
	8: System tells user promotion added

Use Case Diagrams



Functional Requirements

- RQ1: Get Products
- RQ2: Get Promotions
- RQ3: Check Password
- RQ4: Search Products
- RQ5: Filter Products
- RQ6: Get Product
- RQ7: Get Promotion
- RQ8: Get User
- RQ9: Add Product
- RQ10: Add Promotion
- RQ11: Remove Product
- RQ12: UpdateProduct

Traceability Matrix

	RQ	1	2	3	4	5	6	7	8	9	10	11	12
UC													
1		X	X										
2		X	X	X									
3		X	X		X								
4				X					X				
5										X			
6							X					X	
7							X						X
8		X					X				X		

Software System Attributes

Non-functional Requirements

- Scalability

  The system needs to be able to scale to handle multiple users and numerous products as the catalogue grows. A number of businesses currently exist in Hatfield that may want take advantage of such a platform and the system should be able to scale to handle this increase in users and the growth of the catalogue
- Maintainability

  The system needs to be maintainable. The catalogue has to be continuously updated and meet new requirements as well as be able to cope with changing requirements. In addition the system will need to be easy to maintain even after it is in production and for other developers who weren't part of system design.

- Reliability
  The system will have to be reliable and fault-tolerant. The system should not have down-time or be unavailable if an error occurs. The system should be able to handle failures without resulting in downtime. This will be achieved through loosely coupled services that aren't completely dependent on each other to provide functionality.
- Extensibility
  The system needs to be extendable, at this point in time the scope of the system is limited to just browsing but the potential of this platform is so much more. This system needs to be extendable so that other functionality can be added later on.