



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

BELLISIMO  
REQUIREMENTS SPECIFICATION AND DESIGN  
6 AUGUST 2017

---

NOKUTHULA MANANA 12064115

# Contents

<b>Table Of Contents</b>	<b>1</b>
<b>1 Project Background</b>	<b>2</b>
<b>2 Vision and Scope</b>	<b>2</b>
<b>3 Design Specifications</b>	<b>2</b>
<b>4 Interface Designs</b>	<b>2</b>
4.1 Admin Interfaces . . . . .	2
4.2 User Interfaces . . . . .	5
<b>5 Design Technologies</b>	<b>5</b>
<b>6 System Architecture</b>	<b>5</b>
<b>7 Change Factor - User Registration</b>	<b>6</b>
<b>8 Software Metrics</b>	<b>6</b>

# 1 Project Background

Bellissimo is a fictional company aimed at providing an online platform for customers to browse the clothing as well as food catalogues. Information about specials and promotions will be published on this platform. Bellissimo is assumed to have two main focus areas within the business namely, to be a clothing store as well as a supermarket.

## 2 Vision and Scope

The core of the system will be a catalogue of items and their prices. Since Bellissimo is involved in clothing and food, the catalogue will have to ensure that these regions are well maintained. Sales and specials in each region will have to be accounted for and managed. The scope of the system is to ensure that the latest information is being provided about items and their prices.

## 3 Design Specifications

The system should be designed in such a way that the clothing module is mutually exclusive to the food module. That means that the clothing module can be deployed and maintained separately from that of the food module while still maintaining that this information is delivered by one website. The following design specifications are what the system should adhere to:

- There should be at least two interfaces. An admin interface to maintain the catalogue and an anonymous user interface that will allow people to interactively view the catalogue.
- The prices of each item in the database should be displayed along with the image of the item.
- The administrator should be able to add, remove and update any items in the catalogue list.
- The administrator should be able to add specials to the module. Specials can be applied to singular items or groups of items depending on a special group configuration.
- Users should be able to browse, search and filter the catalogue.

## 4 Interface Designs

### 4.1 Admin Interfaces

Below are the prototype designs for the Admin User Interface.

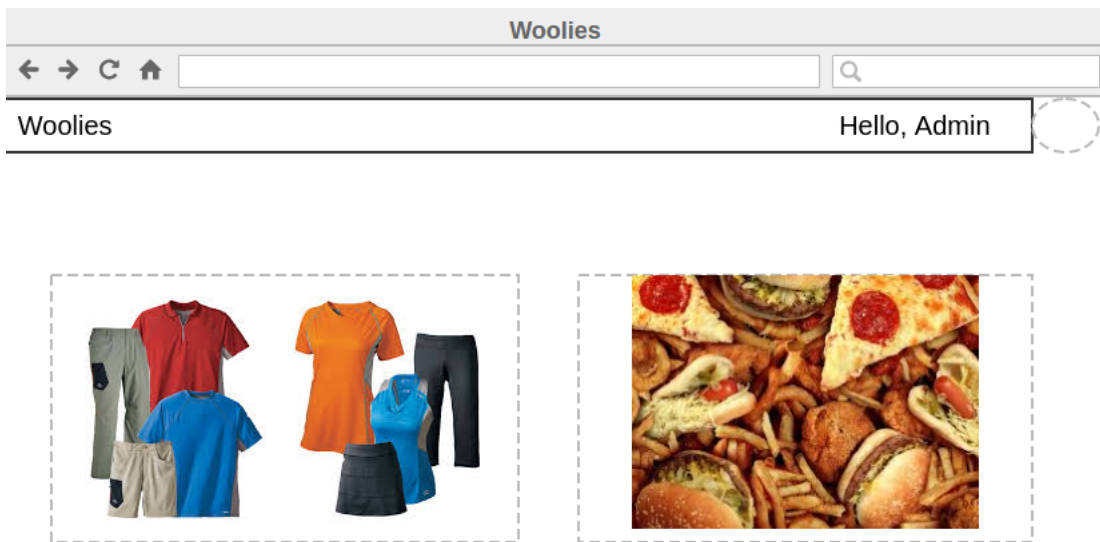


Figure 1: Admin Home Page

Admin has to pick whether they will be dealing with the Food Area or Clothing Area of Bellissimo. Clicking on the relevant image will take them to the correct interface.

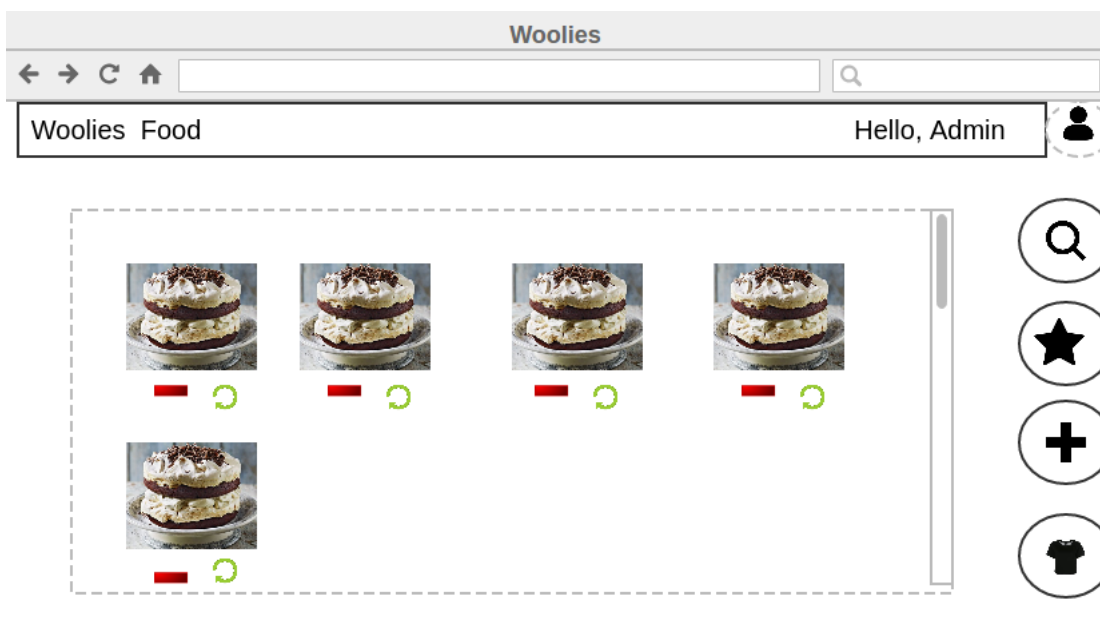
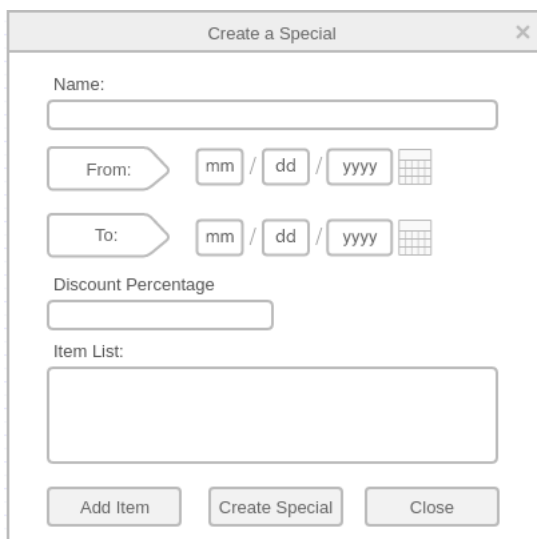


Figure 2: Admin Bellissimo Food Home Page

This is the food homepage where all the items sold by the store are presented in a list. The red minus sign will remove an item from the list when clicked. The green refresh icon will enable admin to make updates to an item, whether it be price, name or simply the image. The four buttons on the side have certain functions as explained below:

- By clicking on **Image 1**, a pop-up search bar will appear on which you can search for a particular item on the list. Searching for the item can be via an item name or id. The item desired, if found will appear in the list of items (along with other items that have the same description).
- By clicking on **Image 2**, a modal will pop-up that will prompt admin to create a special. The *Create Special Modal* will be explained in detail below.

- By clicking on **Image 3**, the modal to add an extra item to the list will pop-up. This modal will be explained in detail below.
- By clicking on **Image 4**, Admin will then be taken to the interface of Bellisimo clothing, which will be explained below.

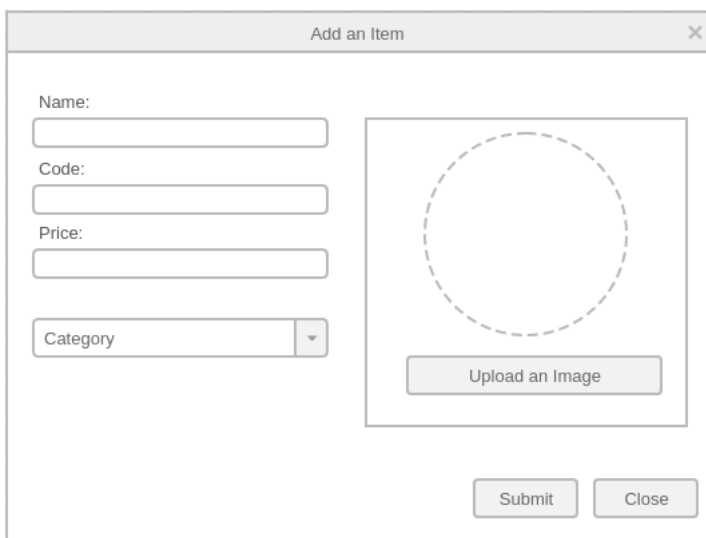


The 'Create a Special' modal contains the following fields and controls:

- Name:** A text input field.
- From:** A date range selector with dropdowns for month (mm), day (dd), and year (yyyy), accompanied by a calendar icon.
- To:** A date range selector with dropdowns for month (mm), day (dd), and year (yyyy), accompanied by a calendar icon.
- Discount Percentage:** A text input field.
- Item List:** A large text area for listing items.
- Buttons:** 'Add Item', 'Create Special', and 'Close' at the bottom.

Figure 3: Admin Bellisimo Create Special Modal

This modal will enable admin to apply a special on a number of items in the repository as selected by Admin. These items must already exist in the database in order to have a special applied on them.



The 'Add an Item' modal contains the following fields and controls:

- Name:** A text input field.
- Code:** A text input field.
- Price:** A text input field.
- Category:** A dropdown menu.
- Image Upload:** A dashed circular placeholder with an 'Upload an Image' button below it.
- Buttons:** 'Submit' and 'Close' at the bottom.

Figure 4: Admin Bellisimo Add Item Modal

This modal allows Admin to add a new item to the repository of items. All the fields in the modal are required except for the image. A default image will be provided if the image is not uploaded.

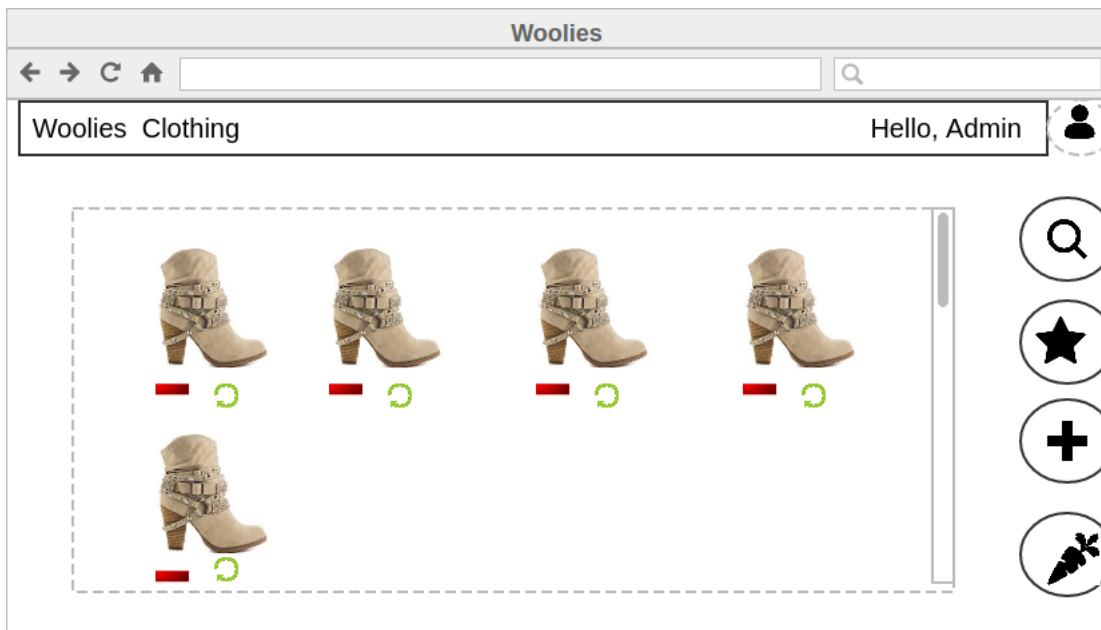


Figure 5: Admin Bellissimo Add Item Modal

The Bellissimo clothing interface is similar to that of Bellissimo Food, with the exception of the heading as well as the carrot icon, which will switch views back to the food interface. The search functionality on this page will be limited to items in the clothing database.

## 4.2 User Interfaces

The user interface is similar to that of the admin with the exception that there are no logins and the *Create Special* and *Add Item* buttons are unavailable.

## 5 Design Technologies

The following technologies will be used to implement the system:

- Html 5 (Html and Bootstrap CSS)
- Angular2
- NodeJS
- Spring Boot
- Spring Cloud (Microservices)
- Netflix OSS (Microservices)
- PostgreSQL
- Apache Maven
- Git (Github)

## 6 System Architecture

The web application will consist of two subsystems that communicate via HTTP using REST Framework. The Java/Spring Boot application will be known as the "backend" application. The HTML5/Angular2 application will be known as the "frontend" application. The backend application is expected to communicate with the database and use Maven as its dependency management tool whereas the frontend application will be hosted in the browser and NodeJS is expected to manage packages required for the application to run successfully.

The backend application will be implemented twice. The first implementation will be under a monolithic architecture while the second will be under a microservices architecture. The frontend application can be used in both the monolithic and microservices versions of the backend application. It will make calls to the frontend application by means of Angular 2's Http module that will target the REST API endpoints of the respective backend applications.

## 7 Change Factor - User Registration

In order to fully test the abilities of microservices architecture a change factor must be introduced to the system once implementation has been completed. This change factor should be significant enough that it can demonstrate the strengths of microservices with this implementation.

The change factor that will be implemented is **user registration**.

A potential client/user should now be able to register on the system. Once registered there should be an interface provided for a registered user. This interface must include the following:

- A list of all that they have purchased on the system (Dating back to 120 days)
- A list of all items that are currently in their cart.
- Information about their profile on the system

Additions to the user interface may be added if desired but these additions must serve a purpose.

## 8 Software Metrics

Software Metrics will then be used to analyse the systems under the different architectures.