UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Bellisimo
Requirement and Design Specification

Diana Obo 13134885

1 Background

1.1 Project Background

Bellisimo is a company aimed at providing an online platform for customers to browse clothing as well as food catalogues provided by the business located in Hatfield. Information about specials and promotions will be published on the online platform.

1.2 Purpose

The purpose of this document is to present the reader with a detailed description of the Bellisimo system. It will delve into the purpose and features of the system, the various interfaces of the system, the capabilities of the system, as well as the constraints under which the system must operate. The content of this document is intended for both the various stakeholders and the developers of the Bellisimo system.

1.3 Visions and Scope

The core of the system will be catalogues of items and their prices. Since Bellisimo is involved in clothing and food, the catalogues will have to ensure that these lines are well maintained. Sales and specials in each line 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. A user should be able to buy items online.

1.4 Architecture Design of Bellisimo System

At the highest level of granularity, the Bellisimo system is based on Monolithic architecture. Second level of granularity can be visualised as to be based on model-view-controller (MVC).

1. Architectural Patterns of the Bellisimo Systems: The architectural patterns of the Bellisimo system are focused and narrowed to patterns. Therefore the following architectural patterns are identified:

(a) Monolithic architecture

(b) Model-view-control architecture patterns

2. Quality Requirements of the Bellisimo Systems: should include but not limited to the following:

   (a) Performance

   (b) Integrability

   (c) Availability - that the latest information about the catalogues are provided

   (d) Maintainability - ensure that the catalogues are well mainainted

   (e) Scalability

   (f) Reliability - sales and specials in each line should be accounted for and managed

   (g) Security - only an admininistrator can access
   the admininistrator interface to create, update and delete food and clothing catalogues.

1.5 Design Requirement

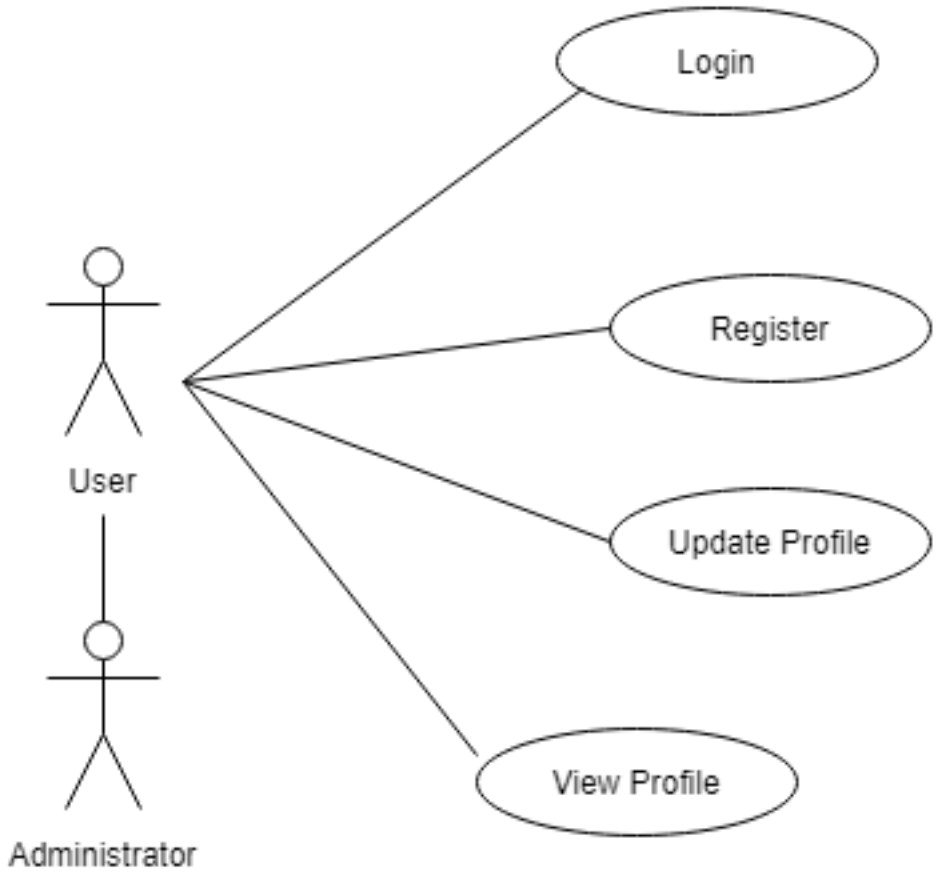
The system should be designed in such a way that there are four different interfaces that depict the clothing catalogue, the food catalogue, user profile and items in cart. The users should be able to register and login to the system. There should be at least two interfaces. An admin interface to maintain the catalogues and a user interface that will allow people to interactively view the catalogues and view, add, remove and update their items in the cart. The prices of each item in the database should be displayed along with the image of the item. A user should also be able to view and update their user profiles.

2 User Management Module

2.1 Scope

The scope of the user management module is shown in Figure 1. The user management module is responsible for maintaining information about the

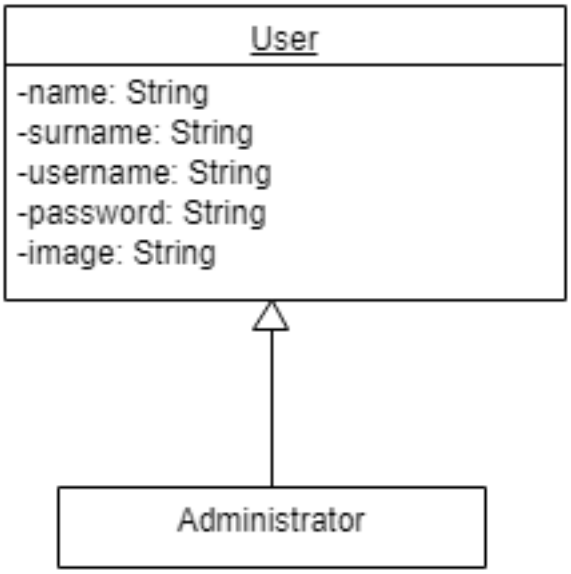
registered users of the system. Administrators can manage information about clothes and food catalogues. A user is able to view and update their user profile.



- User: A user can register and login to the system to gain access to the user interface and once logged in, can view and update their profile.
- Administrator: A default administrator account will be on the database and the administrator need to only login to access the administrator interface. The administrator of the system will receive a guideline pdf on the details of the administrator password and username. The ad-

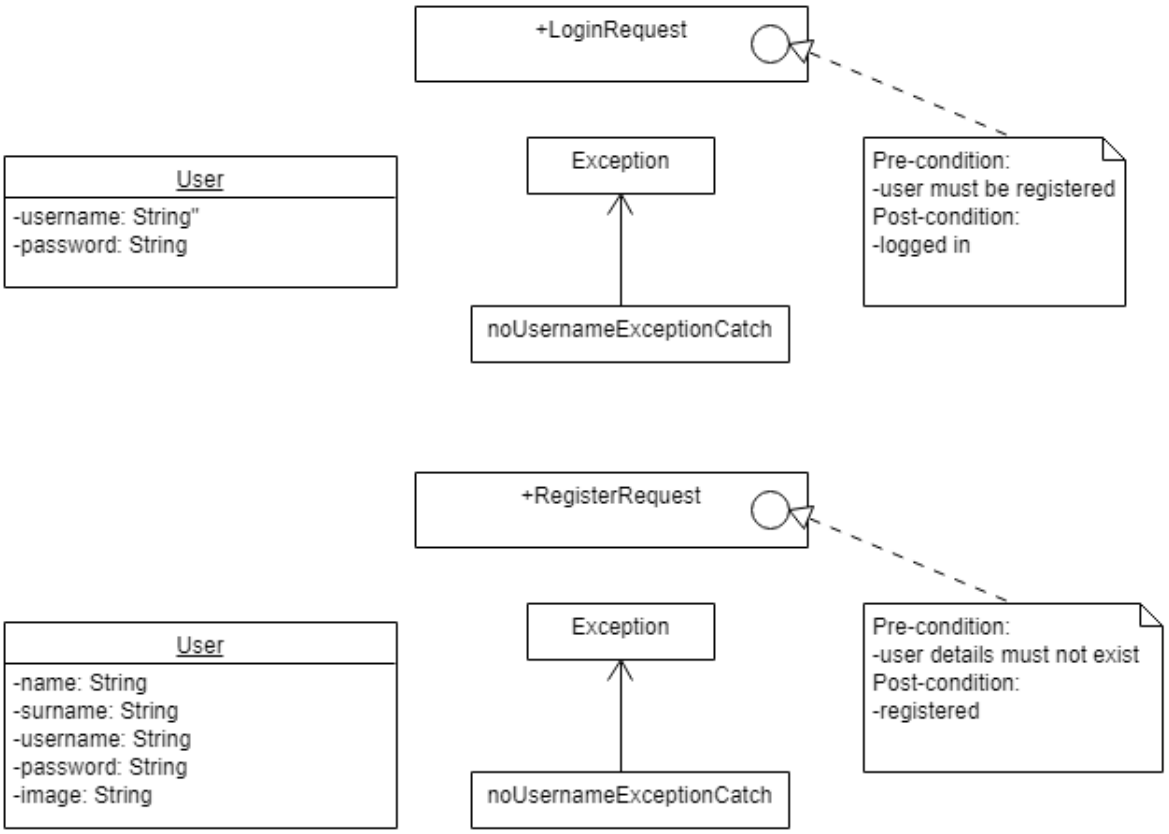
ministrator will be allowed to change their details through the profile interface of the administrator.

2.2 Domain Model



2.3 Service Contracts

- The following services should be provided Precondition: username is a registered user. Exception: If username is not a registered user throw noUsername exception. Postcondition: No change
- The following services should be provided Precondition: username must not exist. Exception: If username exists throw usernameExists exception. Postcondition: No change

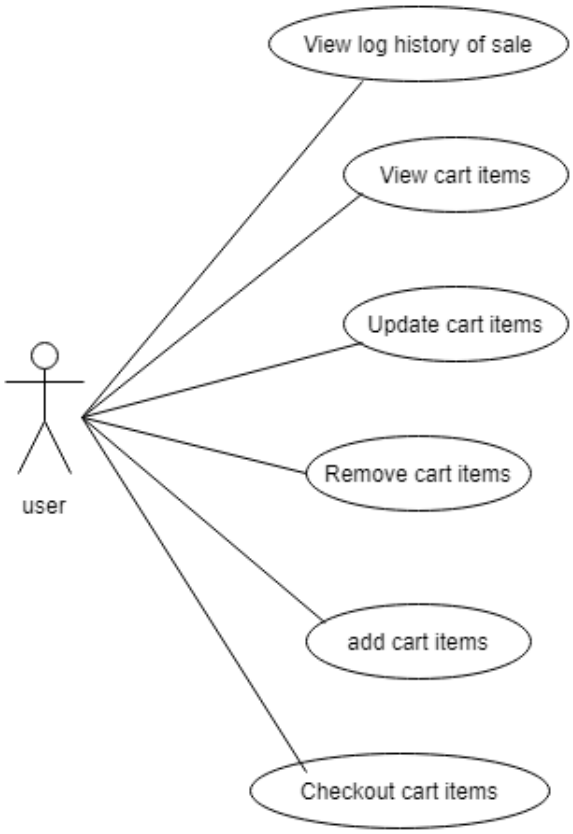


3 Shopping Cart Module

NOT IMPLEMENTED

3.1 Scope

The shopping cart module will manage the log history of alll the sales. the module will allow registered users to view the current items in the cart. The module will allow the registered users to add, remove and update items in the cart. the module will also allow registered users to checkout the items in the cart.



4 Bank management Module

NOT IMPLEMENTED

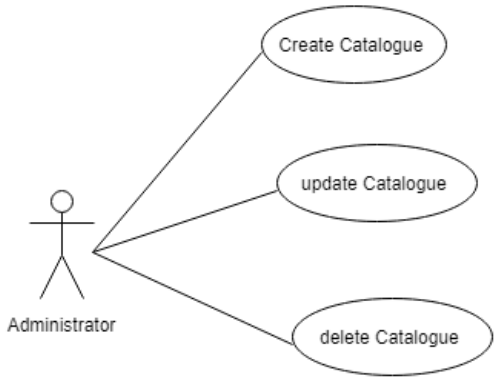
4.1 Scope

The banking management module will use mocking to simulate the role of a bank for electronic payment. The mocking will accept or decline the payment depending on some ratio. The transactions will be persisted if accepted and cancelled if declined.

5 Catalogue Management Module

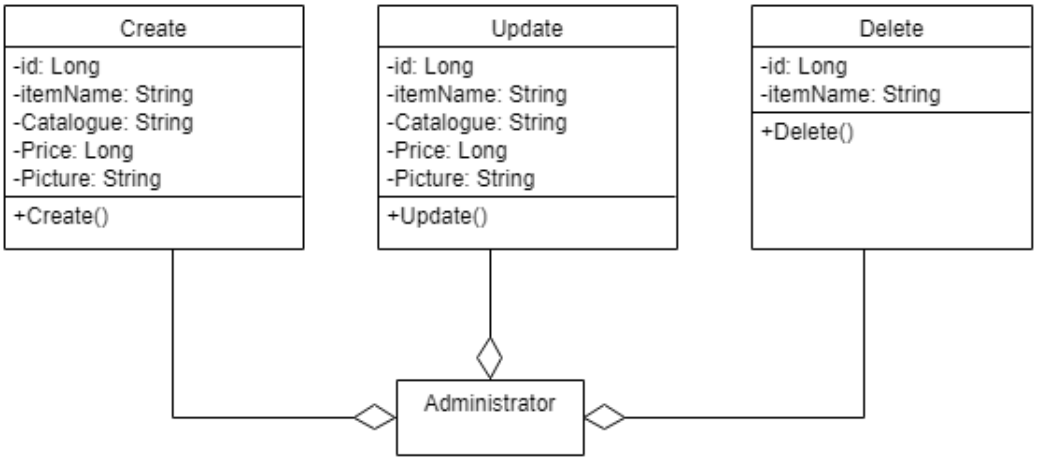
5.1 Scope

The catalogue management module will allow the administrator to add, remove and update any items in the catalogue list. The catalogue management module will also allow the administrator to add specials to the module. Specials can be applied to singular items or groups of items depending on a special group configuration.

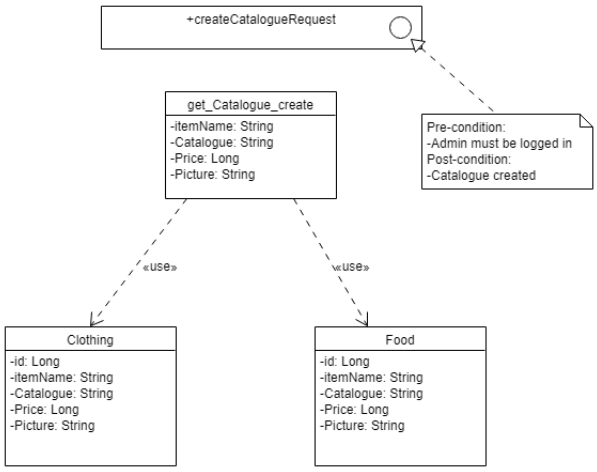


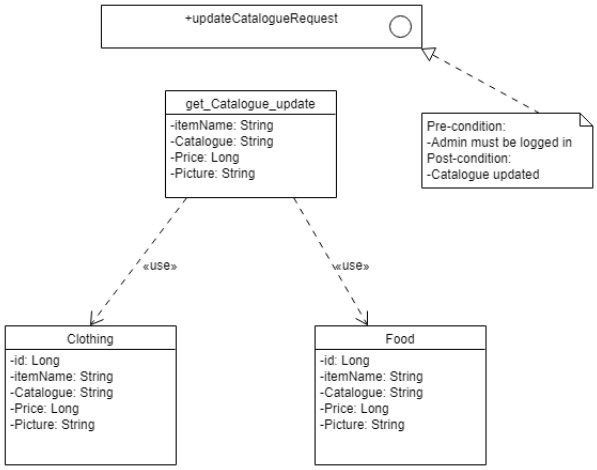
5.2 Domain Model

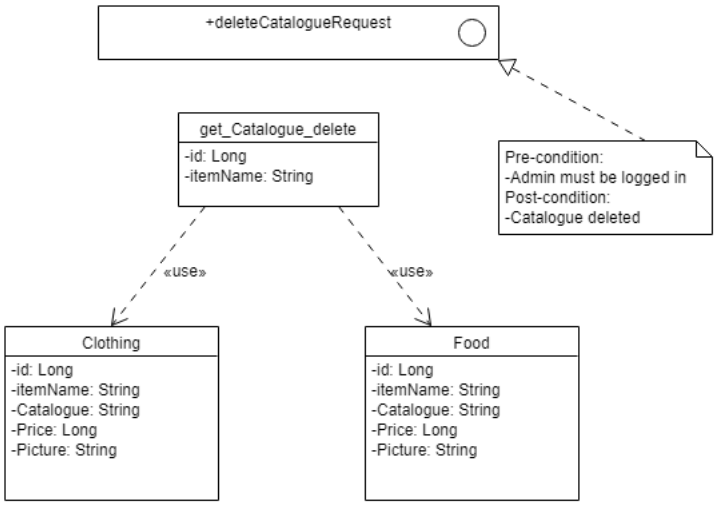
The domain model for the Catalogue Managemen system is simple. It is simply a description how the catalogues can be modified by either being created, being updated or being deleted.



5.3 Service Contracts



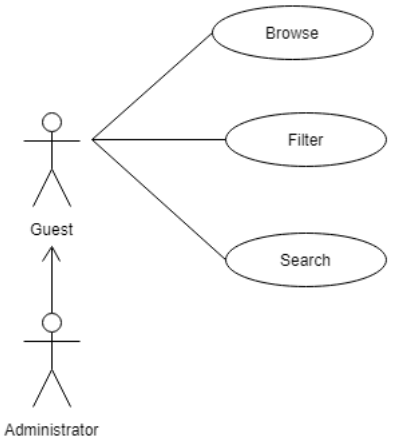




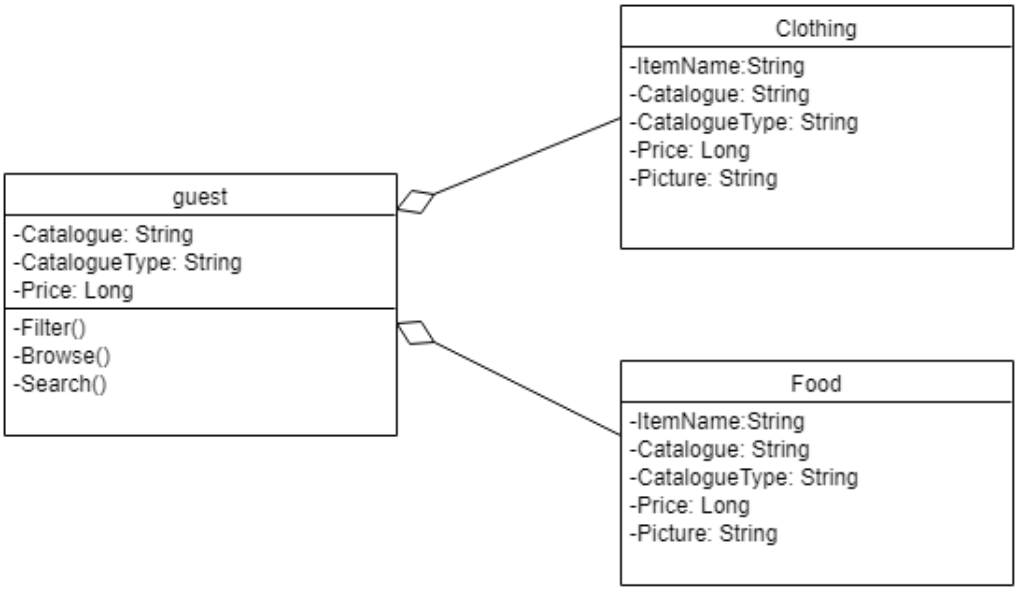
6 View Module

6.1 Scope

The view module will allow the guest user to browse, search and alter the catalogues.

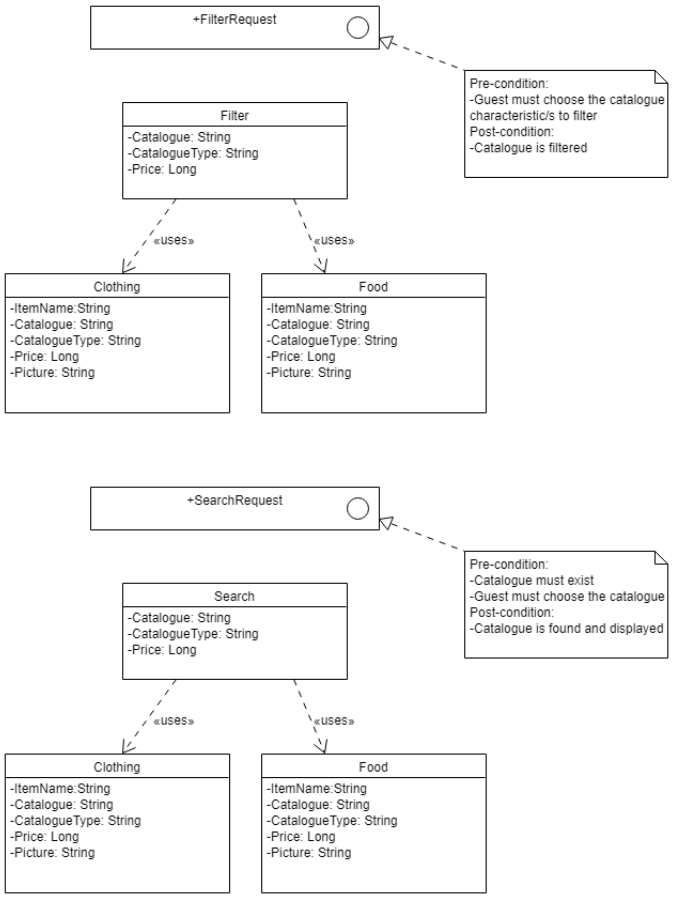


6.2 Domain Model



.

6.3 Service Contracts



7 Architectural Technologies

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 Hibernate which can be imported into Maven, a 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.

7.1 Technologies

The following technologies will be used to implement the system:

- Html 5 (Html and Bootstrap CSS)
- Angular2
- NodeJS
- Spring Boot
- PostgresSQL
- Apache Maven
- Git (Github)

7.2 Monolithic Implementation

The monolithic implementation will consist of one backend spring boot application that will be deployed as a single unit. Apache Maven will be used as the dependency management tool.