UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Department of Computer Science

Bellisimo Solution-Oriented Requirements Specification

Student Name: Mokgadi Maake

Student Number: 13376234

Document version: v0.1

Date 08/09/2017

Table of Contents

List of Tables

List of Figures

DOCUMENT ADMINISTRATION

Terms and Acronyms Used

Table 1 : Terms and Acronyms

Term/ Acronym	Explanation
GOR	Goal-Oriented requirements
SOR	Solution-Oriented Requirements
UML	Unified Modelling Language

1 INTRODUCTION

The purpose of this document is to define the Solution-Oriented requirements (SOR) for the Bellisimo online shopping system. In order to proceed with the SOR, the author will provide a precise description of the Goal-Oriented requirements (GOR) of the system with an intent of making it easier to associate each of the three facets of the SOR with the GOR. The three facets of SOR are named as follows: Data model, functional model and behaviour model. Each of these facets shall be discussed in detail in this document.

2 VISION AND SCOPE

Bellisimo is an online system which is aimed at assisting the users in filtering, browsing, searching and filtering catalogues. Additionally, the system administrator ensures that the line items from clothing and food department are maintained at all times. The administrator of the system shall be able to add, remove and update items on the catalogue. Bellisimo online system will be hosted in the browser and NodeJS is expected to manage packages required for the application to run successfully.

This section defines system context of the Bellisimo system. According, to Pohl (2010) the way the requirement is interpreted is significantly influenced by the documented system context information. Additionally, Pohl (2010) there exist four context facets. Of all these four context facets the Bellisimo system requirement engineer illustrated the usage facet of the Bellisimo using the use case diagrams. The use case diagrams in this context visualise the relationships between the different use cases of the Bellisimo system. The use case diagram illustratively shows the three actors that actively interact with each other. The three actors in discussion are named as follows: The user, the administrator and the Bellisimo system. The objects associated with the Bellisimo system are defined as follows:

- the user and the system administrator are represented using a small stick,
- the Bellisimo system is represented with a box,
- the ellipse represent the use case of the Bellisimo system. The use case is the actions within the box (system) that must be catered for during the development. Each use case has a unique identifier.
- The relationship between actors (user, system administrator, Bellisimo system) and the use case (ellipse) is represented by a line.

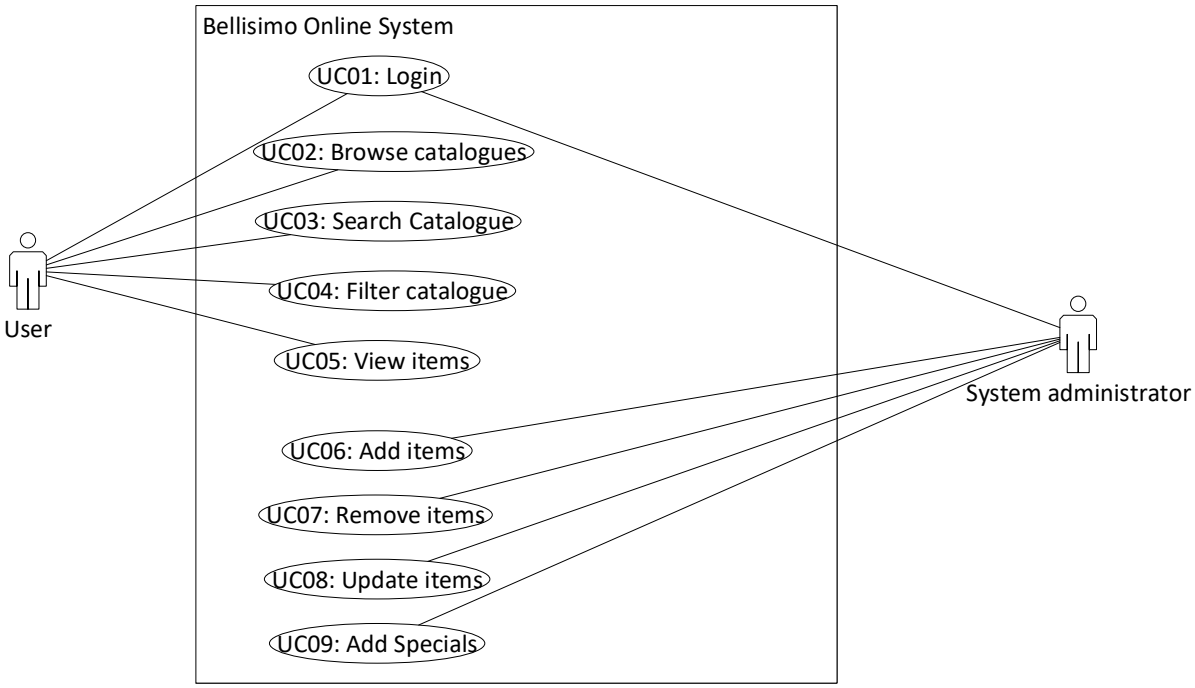

Figure 1: Bellisimo Online System Use Case

2.1 Scenario-oriented requirements

This sections shows how the Bellisimo system interacts with the user and the system administrator. In the context of Bellisimo system, the scenarios define the association between the requirements and usage facet context.

The interaction between the above-mentioned actors is shown by making use of the UML (Unified Modelling Language) that supports the sequence of interactions. In this context of the system Bellisimo system the objects associated are defined as follows:

- the user and the system administrator are represented using a small stick,
- the Bellisimo system is represented with a box,
- the relationship between actors (user, system administrator, Bellisimo system) and the Bellisimo system is shown by an arrow line. The shaded arrow represents the messages from the user and system administrator, whereas the unshaded arrows represent the response messages from the Bellisimo system.
- action box represents the processing of the response messages within the Bellisimo system.

The Bellisimo system interactions are illustrated as follows:

2.1.1 UC01: Login

The sequence diagram that is shown in Figure 1 illustrates goal satisfaction of use case UC01: Login. User login.

A successful login is generation as is shown by the dotted arrow from Bellismo system pointing to the system administrator in the sequence diagram shown in Figure 2.

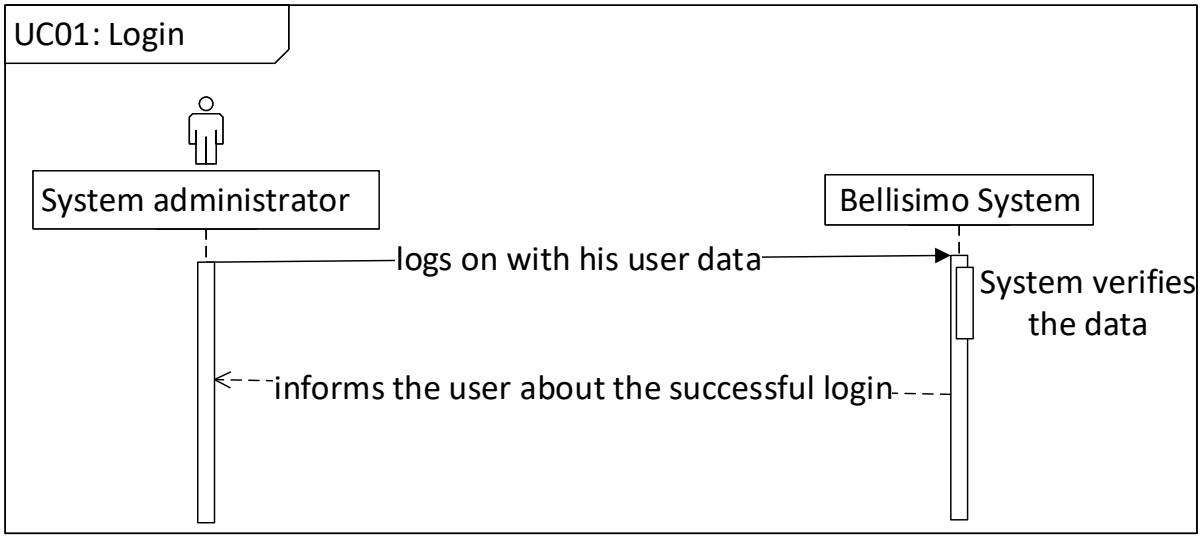

Figure 2 : UC01: Login

2.1.2 UC02: Browse catalogues

This sequence diagram in Figure 3 illustrates goal satisfaction of use case UC02: Browse catalogues.

The system user opens the Bellisimo webpage and browse through the items. The system retrieves the items with their associated images from the database and displays to the user.

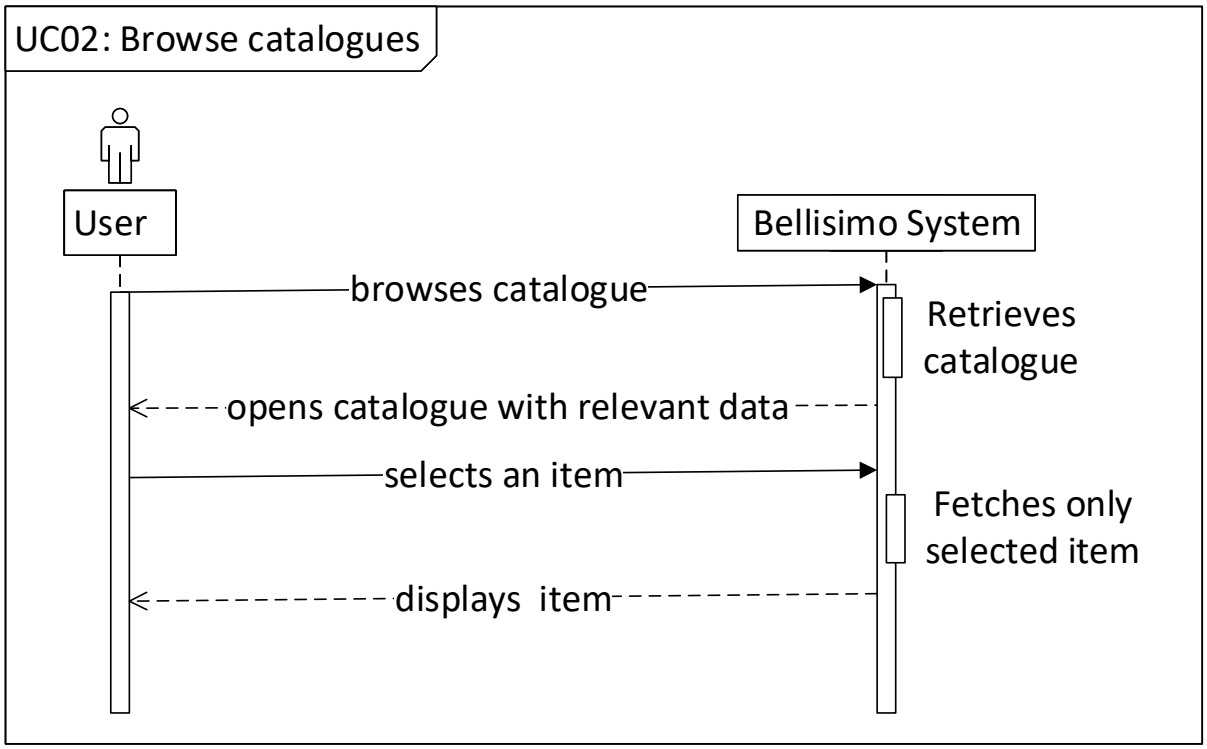

Figure 3 : UC02: Browse catalogues

2.1.3 UC03: Search Catalogue

The sequence diagram in Figure 4 illustrates goal satisfaction of use case UC03: Search catalogues. On the web page the user enters the items they desire and click on the search button. Therefore, Bellisimo system fetches the items from the database and display to the user accordingly.

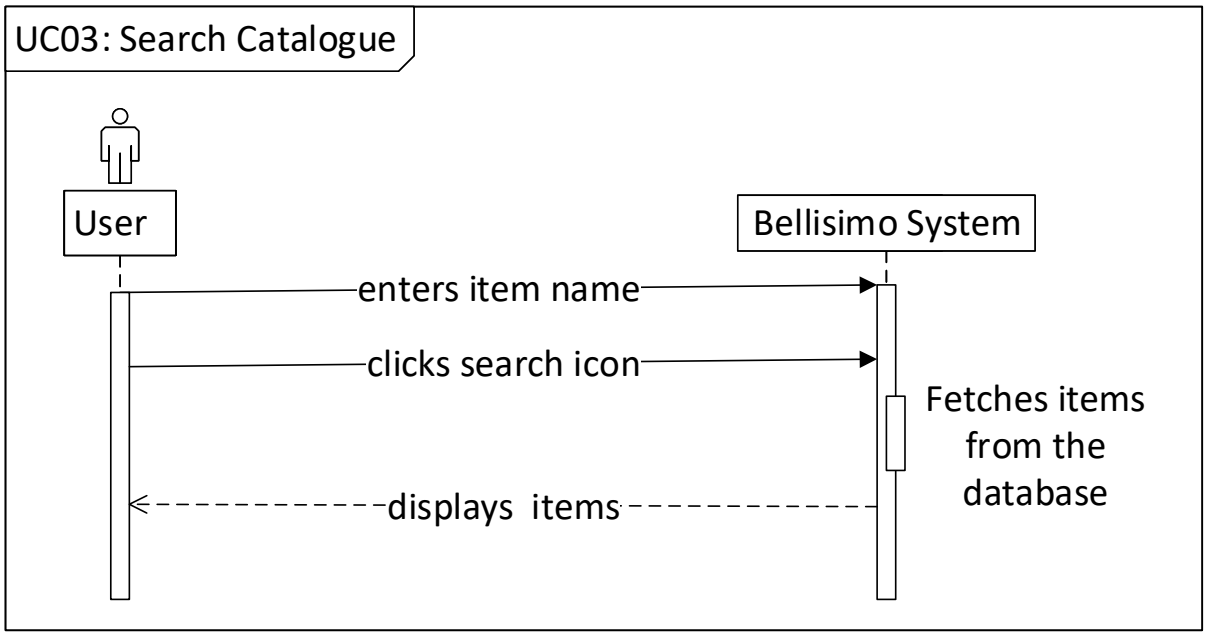

Figure 4 : UC03: Search catalogue

2.1.4 UC04: Filter catalogue

This sequence diagram in Figure 5 illustrates goal satisfaction of use case UC04: Filter catalogue.

The user filters items using the commodity type button e.g. dairy, fruits, etc. Alternatively, the user filters items at the department level e.g. clothing, food.

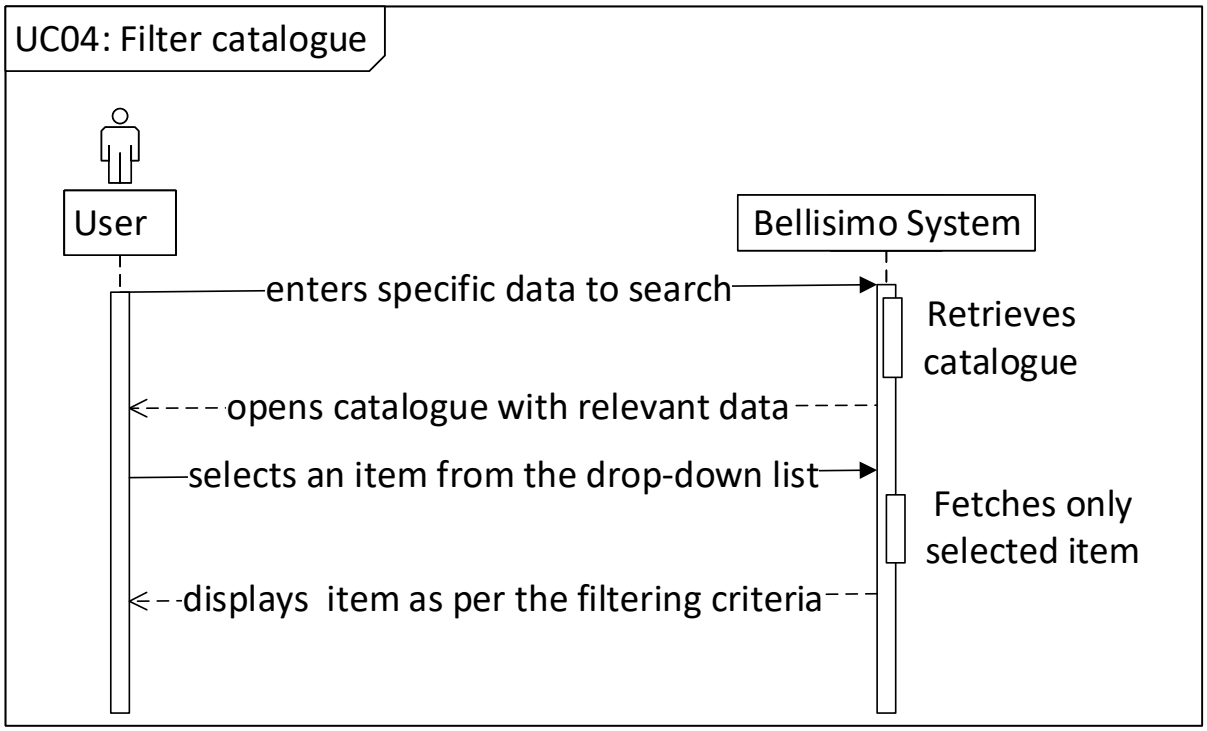

Figure 5 : UC04: Filter catalogue

2.1.5 UC05: View items

This sequence diagram in Figure 6 illustrates goal satisfaction of use case UC04: Filter catalogue. The user selects an item by clicking on it and the Bellisimo system displays the item with its image. Additionally, the item will have special offer attached to it.

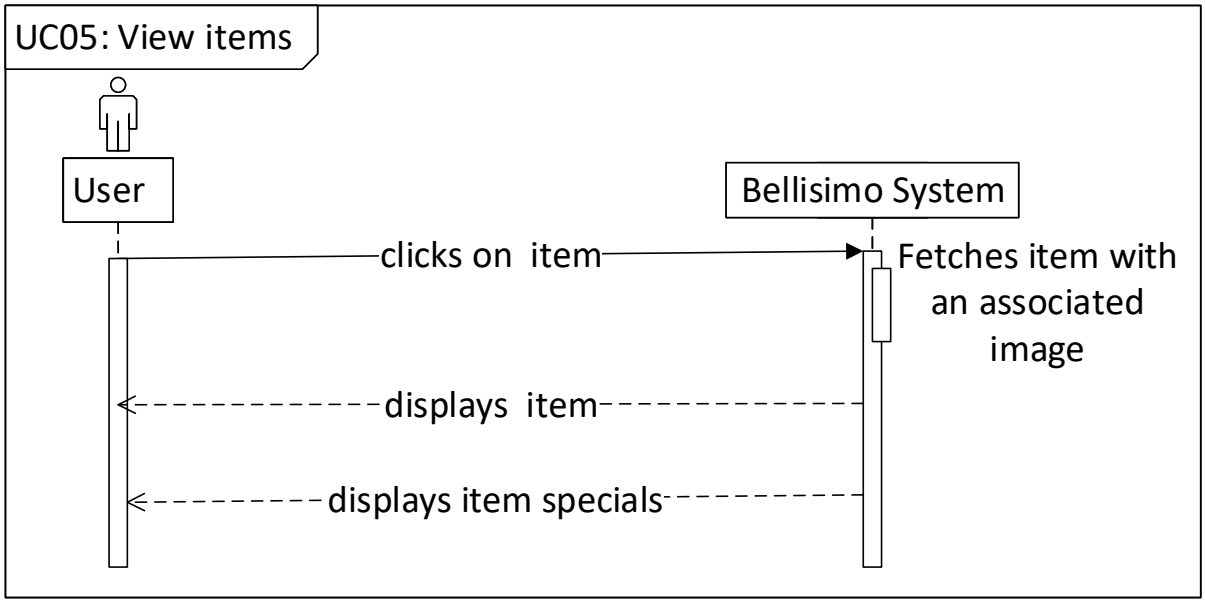

Figure 6 : UC05: View Items

2.1.6 UC06: Add items

The sequence diagram shown in Figure 7 illustrates goal satisfaction of use case UC06: Add items. The system administrator selects the commodity type of an item or department on the Bellisimo system dashboard. The system activates the add functionality to enable the administrator to add the unique identifier items. The system has a set threshold, if the maximum limit has reached, the system reject new items and if the below threshold the items will be uploaded successfully.

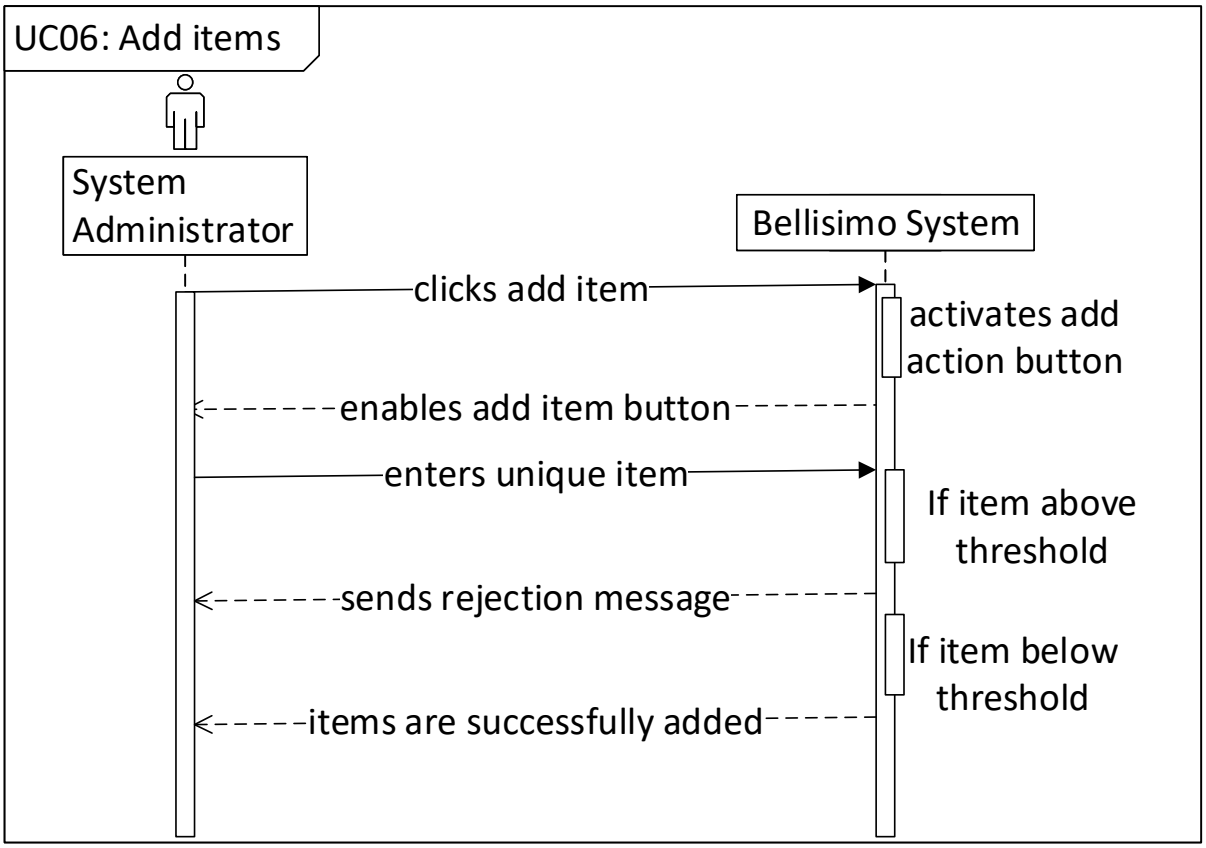

Figure 7 : UC06: Add items

2.1.7 UC07: Remove items

The sequence diagram shown in Figure 8 illustrates goal satisfaction of use case UC07: Remove item. The system administrator selects the commodity type of an item or department. The system activates options to add, remove, update or add special to the item. The system administrator select remove option and enters unique item, then click remove on the Bellisimo system. The system displays the message that the item has been removed successfully.

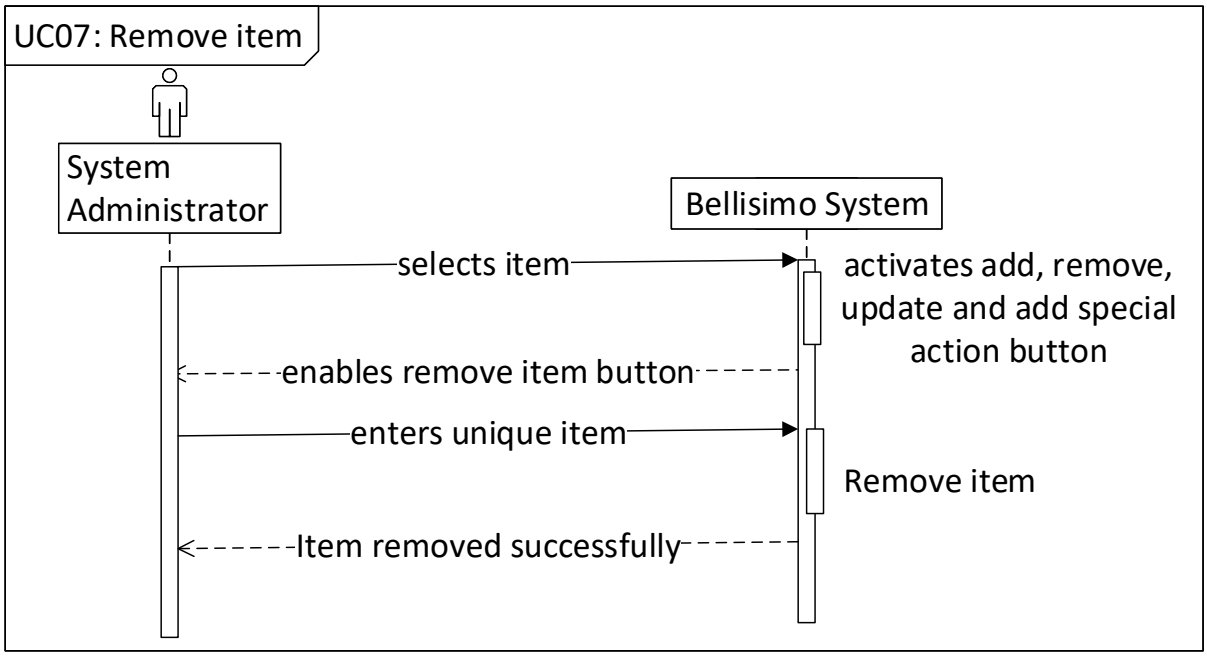

Figure 8 : UC07: Remove items

2.1.8 UC08: Update items

This sequence diagram that is shown in Figure 9 illustrates goal satisfaction of use case UC08: Update items. The system administrator selects the commodity type of an item or department. The system activates options to add, remove, update or add special to the item. The system administrator then selects update option and enters unique item, then click update on the Bellisimo system. The system displays the message that the item has been updated successfully.

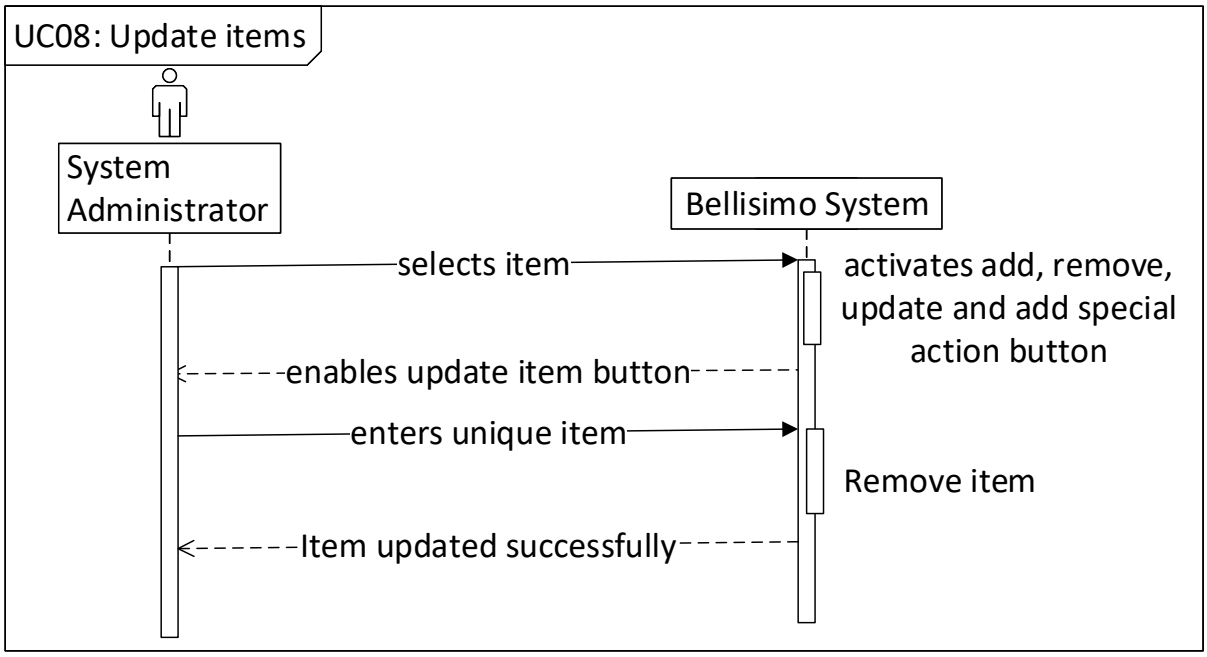

Figure 9 : UC08: Update items

2.1.9 UC09: Add specials

This sequence diagram that is shown in Figure 10 illustrates goal satisfaction of use case UC06: Add specials. The system administrator selects the commodity type of items on special. The system activates options to add, remove, update or add special to the commodity type. The system administrator select update option on the unique item, then clicks update. The system displays the message that the item has been updated successfully.

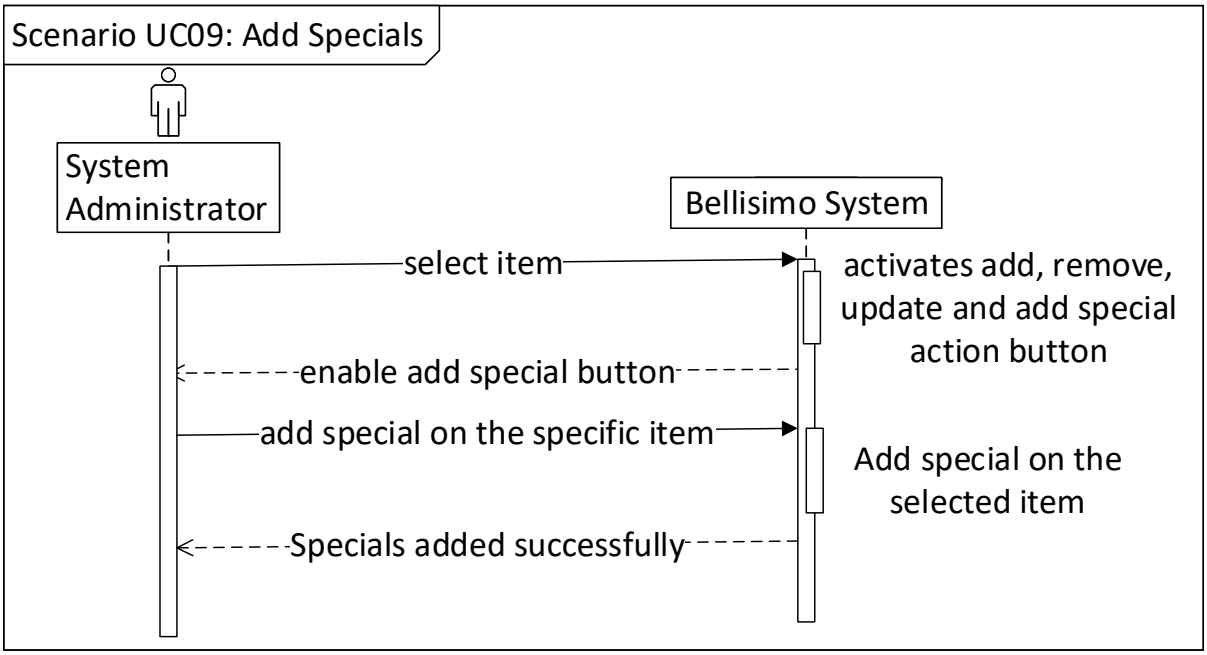

figure 10 : UC09: add specials

3 GOAL-ORIENTED REQUIREMENTS

This section goes further to give a brief discussion on the business requirements of the Bellisimo online system. The section further defines the Bellisimo Requirements Specification document dated 10 July 2017 provided by Dr. Vreda Pieterse. Each requirement's intention is precisely documented and the scenarios associated with the requirement. The scenarios can be easily traced in Table 2.

Table 2 : Goal-oriented requirements

Identifier	Name	Source	Using Stakeholder	Goal Description	Main scenarios
G1	Interfaces	Bellisimo Requirements Specification document 10 July 2017.	System Administrator / User	The system shall have two intuitive interfaces to allow the system administrator and users	In order to fulfil this requirement, see UC01: Login.

				to be able to login.	
G2	Browse catalogues	Bellisimo Requirements Specification document 10 July 2017.	User	The system shall display catalogue for clothing and food.	In order to fulfil this requirement, see UC02: Browse catalogues.
G3	Search catalogue	Bellisimo Requirements Specification document 10 July 2017.	User	The system shall allow the user to search catalogue.	In order to fulfil this requirement, see UC03: Search catalogue.
G4	Filter catalogue	Bellisimo Requirements Specification document 10 July 2017.	User	The system shall allow the user to filter catalogue.	In order to fulfil this requirement, see UC04: Filter catalogue.
G5	View items	Bellisimo Requirements Specification document 10 July 2017.	User	The system shall display the item with an image and specials.	fulfil this requirement, see UC05: Filter catalogue; UC05.01: Item specials.
G6	Add item list	Bellisimo Requirements Specification document 10 July 2017.	System Administrator	The Bellisimo online system shall allow the system administrator to add items in the catalogue list.	In order to fulfil this requirement, see UC06: Add items.
G7	Remove item list	Bellisimo Requirements Specification document	System Administrator	The Bellisimo online system shall allow the	In order to fulfil this requirement, see UC07:

		10 July 2017.		system administrator to remove items in the catalogue list.	Remove items.
G8	Update item list	Bellisimo Requirements Specification document 10 July 2017.	System Administrator	The Bellisimo online system shall allow the System administrator to update items in the catalogue list.	In order to fulfil this requirement, see UC08: Update items.
G9	Add module specials	Bellisimo Requirements Specification document 10 July 2017.	System Administrator	The Bellisimo online system shall allow the system administrator to add specials to the module.	In order to fulfil this requirement, see UC09: Add module specials.

4 SOLUTION-ORIENTED REQUIREMENTS

This section discusses model-based documentation of the solution-oriented requirements in three perspectives namely Data model, Functional model and Behavioural model.

4.1 Data perspective

This section defines the data that is managed by Bellisimo system. In the context of Bellisimo system the following objects were used to represent data:

- Class which is rectangle in shape with three sections (name, attributes, and operations performed by the class).
- Attributes which are the data types within the class object.
- Associations –which describes the relationships between the classes. This object is represented by a shaded line. The multiplicity between the classes is also specified.
- Aggregation relationship – specified by a whole-part arrow with a shaded heart. This means that there exist one owner and cannot exist independently.
- Composition -

The UML class diagram in Figure 11 specifies the data model of the Bellisimo system. The classes are represented as follows:

- Password class has one to one association with the Base user. This class model relates to use case UC01 Login.
- UserId class has one to one association with the Base user. This class model relates to use case UC01 Login.
- Base user is an abstract class.
- Administrator has an aggregation relationship with the Bas User class.
- User has an aggregation relation with the Base User class. This class relates to Relates to UC02: Browse catalogues, UC03: Search Catalogue, UC04: Filter catalogue and UC05: View items
- ItemImage class contains images that will be utilized in the item Class. This responds to UC05: view items.
- Item class contains item description attributes. This class calls ItemImage to obtain images of the Items.
- Commodity class has a composition relation with Item class.

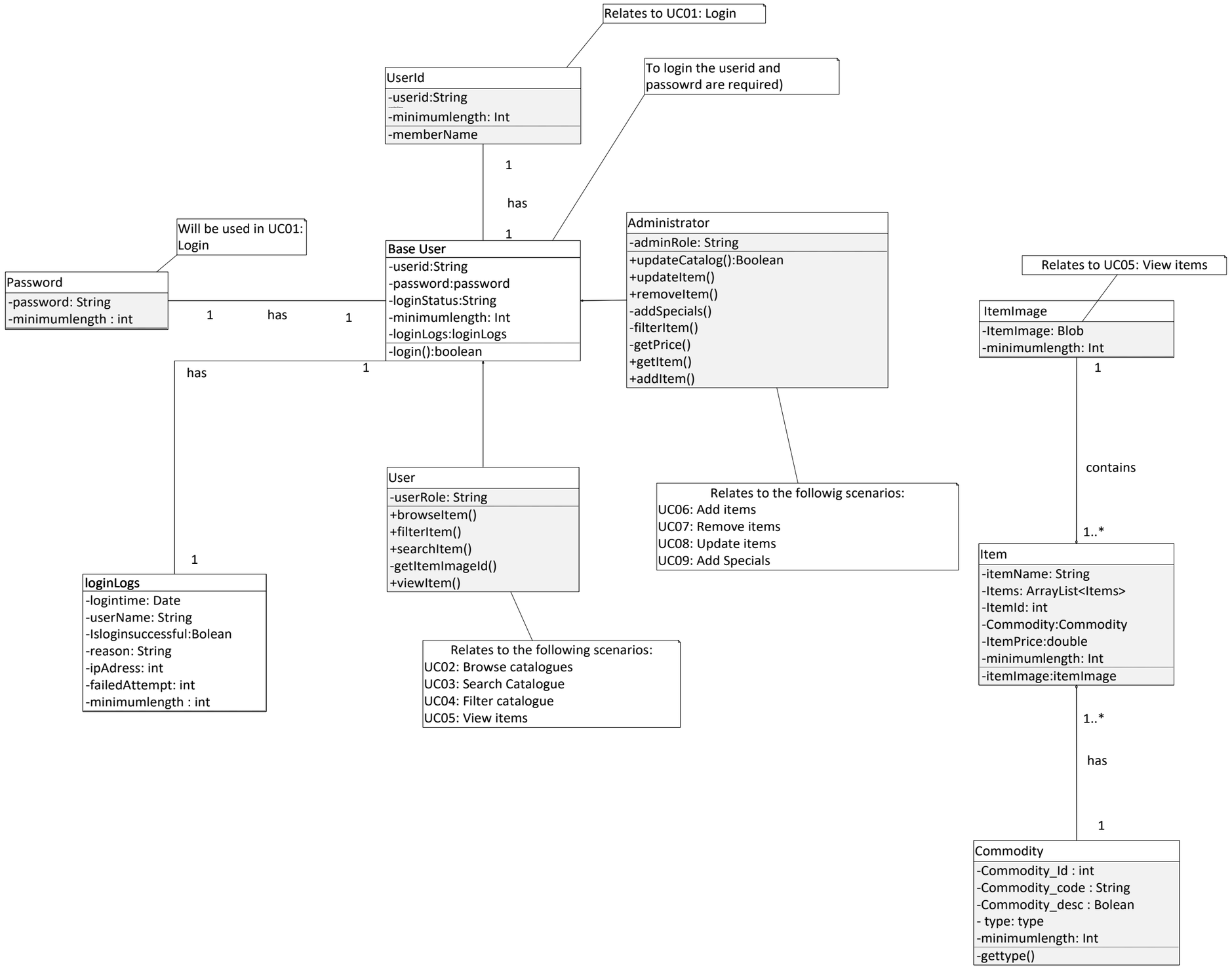

Figure 11 : Class Diagram

4.1.1 Assumptions

The following assumptions are made for the class diagram in figure 11.

- The Department and types are manually maintained by running the scripts against the Postgres database via database clients.
- The password is not hashed on the database.
- The first admin user will be added manually using a script to insert into the user profile (name, surname, password, role).
- The primary key ID Is auto generated in all tables.
- Roles are hard-coded on the back-end layer via Spring-boot

4.2 Functional perspective

This section illustrates the functional dependencies of the processes which are the input data and output data. In the context of Bellisimo system the following objects were used to functional model of the Bellisimo system:

- the cycle objects represent the activity that is implemented in the Bellisimo system.
- the rectangle small box represents the sources that communicates with the system.
- the arrows represent the data movement.
- the large square box groups the functions into different levels

Figure 12 gives a contextual representation of the functional perspective of the Bellisimo system.

The following defines the activities as shown on data flow diagram:

- S1: User Id and S2: password is the input into the Bellisimo system.
- P1 : verifies login details. This is an activity performed by the Bellisimo system.
- S3: Login logs, S4: Item, S5: commodity is output of the process. S3 and S5 serve an input to P2. P2 process the activities (P2.1, P2.2, P2.3, P2.4, P2.5, and P2.7,) to output S3: Item.

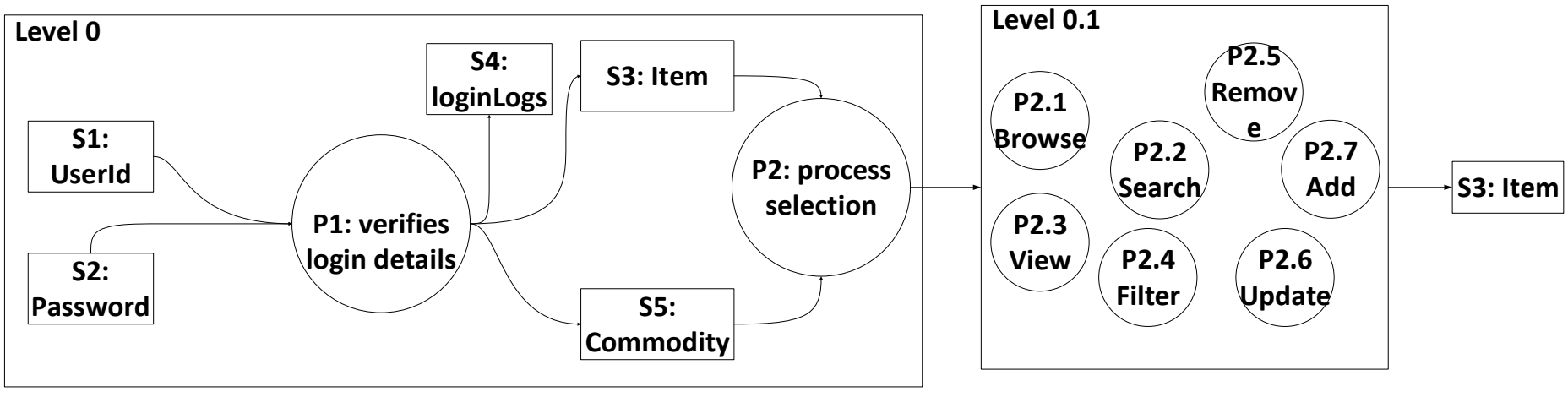

Figure 12 : Data Flow Diagram

4.3 Behavioural perspective

This section defines the behaviour of the Bellisimo system. It specifies how the activities are done and how they happen using objects. The following objects were used to represent how the activities are achieved:

- A round circle represents the start of an activity.
- The state is represented by a square object, which represents an action performed within the system.
- Transition is represented by a line which indicates flow of information.
- A diamond object represents the decision point.

4.3.1 Behaviour specification 1

Behaviour specification relates to the functional activity in P1. The navigation of the system is shown using the state machine diagrams.

- State “connect to the Bellisimo database”: In this state, the system administrator enters log in user id and password. The information is passed to the database to determine if the login details are valid.
- State “Output items”: The system processes the user login details. If the login details are correct, the system will output the items. Therefore, the system administrator can therefore add, delete, update and remove items.
- State “Output error message”: The system processes the user id and password. If not correct the system will display an error message.

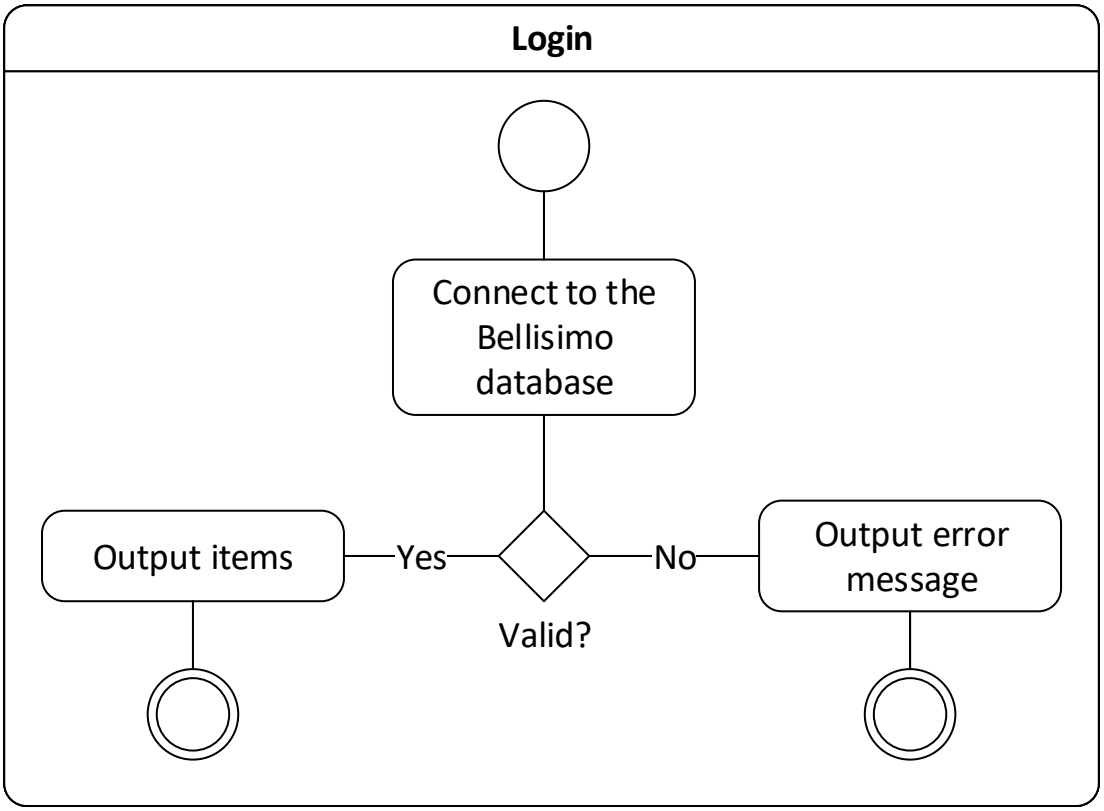

Figure 13 : State Machine Diagram Behaviour 1

4.3.2 Behaviour specification 2

Behaviour specification relates to the functional activity in P2. The navigation of the system is shown using the state machine diagrams.

- State “Invoke filtering”: In this state, the user clicks on the bottom page to filter items.
- State “Invoke search/browser”: In this state, the user clicks on the middle of the page to browse or search items.
- State “Invoke view”:” In this state, the user clicks on the top of the page to view items.

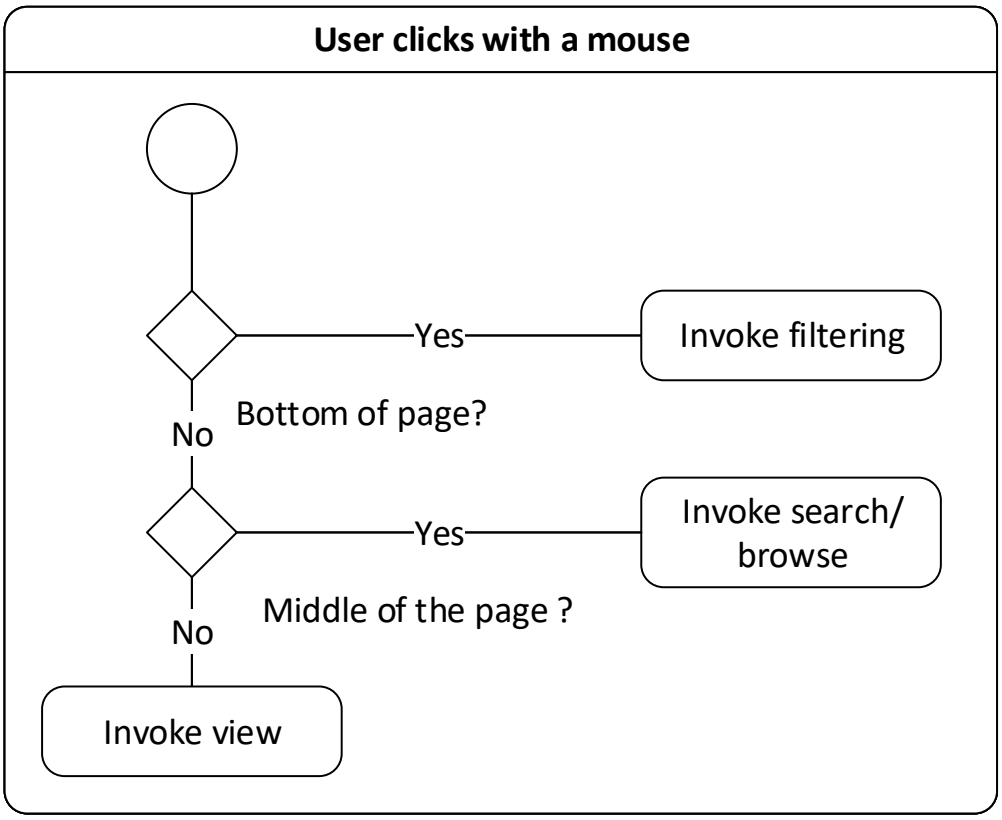

Figure 14 : State Machine Diagram Behaviour 2

5 BELLISIMO SYSTEM ARCHITECTURE REQUIREMENTS AND DESIGN

This section defines the requirements that direct the application architecture design for the Bellisimo system. The monolithic architecture was carefully chosen for the development of Bellisimo system mainly based on its proven traditional benefits of web applications. Over and above, the main advantage of monolithic is that the business logic for servicing the user request is packaged into a single component. The other advantage of monolithic is its scalability. Figure 14, depicts the high-level overview of the micro-services architecture for the Bellisimo system.

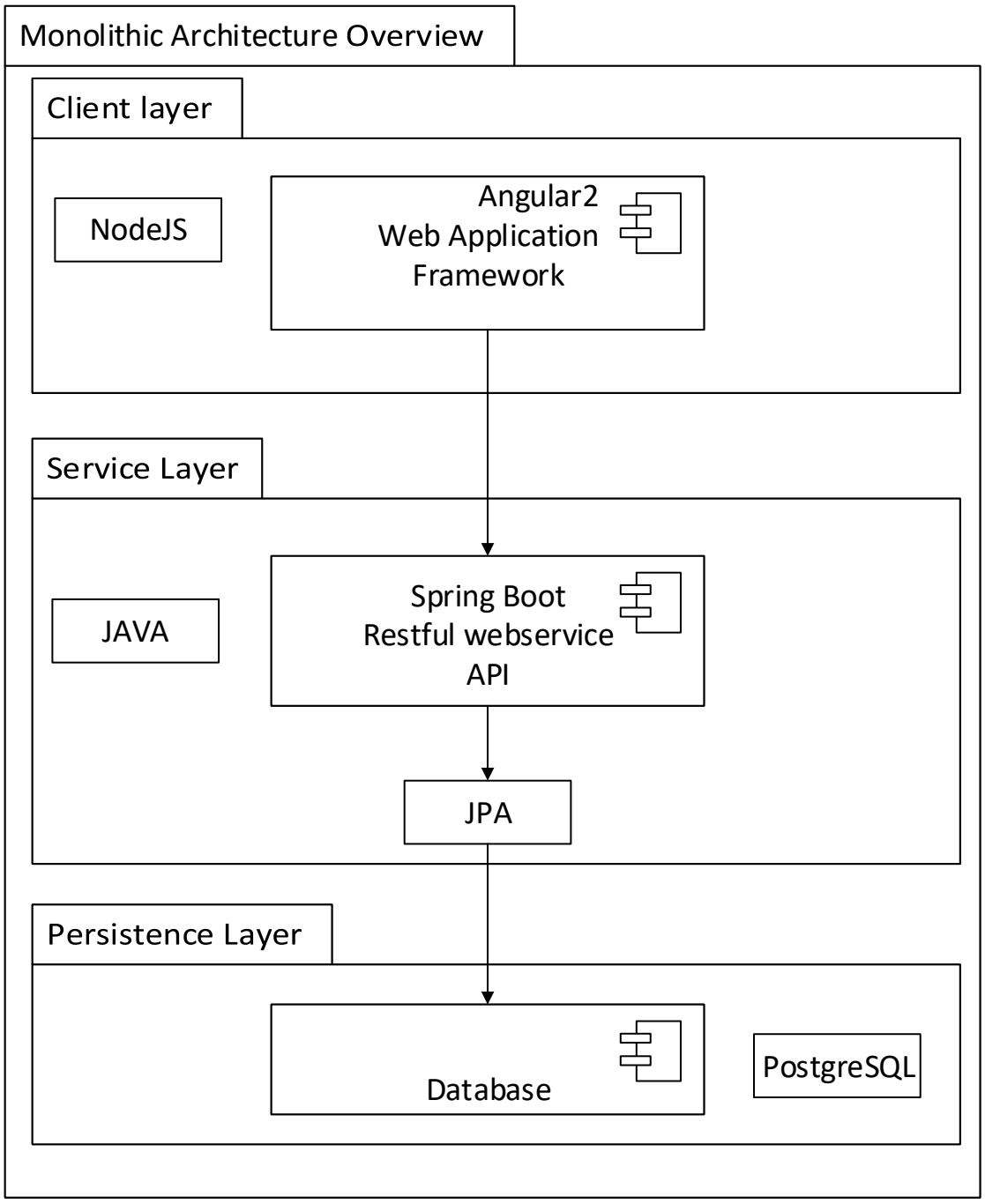

Figure 15 : Micro-services Architecture Overview

5.1 Architecture requirements

This section defines the quality requirements associated with the Bellisimo system. This requirement includes Flexibility, maintainability, testability, usability, portability.

5.1.1 Flexibility

Based on the architectural technologies used, Bellisimo system satisfies this requirement. The application can be easily extendible with new functionalities.

5.1.2 Maintainability

The technologies used for the Bellisimo system are still new in the e-commerce sectors. The system administrator can easily add new items, remove items and add new functionalities into the system.

5.1.3 Testability

Each component of the Bellisimo system can be tested independently without having an impact on other components of the system.

5.1.4 Usability

The Bellisimo system is intuitive. The system is built on assumptions that users are computer knowledgeable.

5.1.5 Portability

The Bellisimo system is built on scripts only and easily deployable. The back-end Spring boot is running on tomcat runtime. The front-end Angular is running using JIT (Just in time runtime).

5.1.6 Security

Only authenticated system administrator is allowed to delete, add special, update items and remove items.

6 REFERENCES

Pohl, K. (2010). *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated.