|  |
| --- |
| **UNIVERSITY “POLITEHNICA” OF BUCHAREST** |
| **SOFTWARE REQUIREMENTS SPECIFICATION**  **“MusicDepot” Java Based application**  *Student:*  *Cristina GEORGESCU- MSE1*  *Coordinator: Date created:*  *Prof. Andrei VASILATEANU Monday, December 3rd, 2018*  ***Table of contents***  [***1. Introduction*** *- 3 -*](#_Toc534561781)  [*1.1. Purpose - 3 -*](#_Toc534561782)  [*1.2. History - 3 -*](#_Toc534561783)  [*1.3. Scope - 3 -*](#_Toc534561784)  [*1.4. Definitions, Acronyms and Abbreviations - 3 -*](#_Toc534561785)  [*1.5. Structure - 3 -*](#_Toc534561786)  [***2. General description*** *- 4 -*](#_Toc534561787)  [*2.1. Product description - 4 -*](#_Toc534561788)  [*2.2. Product functions - 4 -*](#_Toc534561789)  [*2.3. User description - 5 -*](#_Toc534561790)  [*2.5. Assumptions and dependencies - 5 -*](#_Toc534561791)  [***3. Specific requirements*** *- 5 -*](#_Toc534561792)  [*3.2. Performance requirements - 7 -*](#_Toc534561793)  [*3.3. Interface requirements - 7 -*](#_Toc534561794)  [*3.4. Software system attributes - 7 -*](#_Toc534561795)  [***4. Non-functional requirements*** *- 10 -*](#_Toc534561796) |

**“MusicDepot” - Online Shopping Cart for a Music Store**

# Requirements Analysis

According to IEEE STD-830-1993, IEEE Recommended Practice for Software Requirements Specification.

# 1. Introduction

## **Purpose**

The purpose of this document is to present the software requirements specification of the Java-based project *“MusicDepot”.*

## **History**

The first draft was made on December 3rd, 2018.

## **Scope**

The “*MusicDepot*” application is intended to be used by any random user that wants to buy one or more music albums, choosing from a diverse range of music genders.

This project represents a simple Java replication of an Online Music Store. It is aimed to combine the properties of 3 programming paradigms into a cohesive, applicable-in-real life situation. For the Imperative/Functional part, the coding is done with Java in Eclipse and the Declarative part, i.e. the rules that the structure of the program should follow, is done in Drools.

## **Definitions, Acronyms and Abbreviations**

## **Structure**

Chapter 2 provides the general description of the “*MusicDepot”* project*:*

* product description and functions
* user description
* constraints
* assumptions
* and dependencies

Chapter 3 presents the system requirements:

* external interface
* functional, and performance requirements
* design constraints
* software system attributes
* use case diagrams

Chapter 4 presents the Non-functional requirements

# 2. General description

## **2.1. Product description**

“*MusicDepot*” is a common and simple to use, Java application, created for the everyday person. The target audience is the general population. The application is very user friendly and easy to use by any category of age. There is also a wide variety of searches that can be performed.

Being a Java-based application, it can be run on any device with Java JDK installed. Considering this low system requirement, it allows the clients to access the application from any device, with no specific requirement of Internet access that is connected to the internet.

There are no complicated procedures, there is a high compatibility with the majority of devices on the market.

The application has a target audience of any user that is interested in accessing the online store, with no regards of their technological background.

The application is built to be as user-friendly as possible, simple, with guidance messages where needed and good display.

## **2.2. Product functions**

The main participant is the User itself that will do all the actions in the User Interface.

The products are written in a separate Excel file and imported into Java.

Every product is described the following properties:

* Title, that includes the artist, album name, year of appearance on the market
* Genre
* Price (in €)
* Quantity available (the user will not see this number)

Eg: (“Travis Scott, Astroworld”, 2017, “hip-hop”, 20.95, 1500) -- as in the order from above

The User Interface is the first window that the user will see. This interface is divided into several parts, each with its own functionality:

* The section where the available products are visible on the screen (the products are displayed as a scroll down list)
* The “Search” button, that allows the user to search all/part of the title of a product
* The “Add to Cart” button, right next to which the user can enter the desired quantity of a selected product
* The “Reset Cart” button, that erases all the items added up till that point
* The “Checkout” button that will take the user to a new page

In this new page the user will get to see the list of the selected items, the quantities entered previously, price for each item, and, at the very end of the list, the Grand Total to pay. There is a shipping fee of 10 € added to the Grand Total. In this stage there are some rules that apply:

* If a selected item is out of stock
* If the quantity desired is greater that the available quantity at that moment

For these 2 rules, there is a “Verify items” button, which will check and update the item list by displaying the status for each product.

* If the user has a coupon – of type GMC10, there will be a 10% discount applied to the Ground total
* If the user enters his/her e-mail address, to subscribe to the newsletter, then the shipping fee is subtracted from the Grand Total

## **2.3. User description**

The “*MusicDepot*” application is designed any type of user, regardless of age, technical background or technical skills. Everything is done in a user-friendly mode, as straight-forward as possible. There is no Log In option.

**2.4. Constraints**

“*MusicDepot*” requires the previous installation of a Java IDK, Eclipse/NetBeans/any other program that can read and run Java code, as well as Drools rule engine with tools and the Maven libraries.

## **2.5. Assumptions and dependencies**

The applications can run on several operating systems, as long as the constrains mentioned above are respected.

# 3. Specific requirements

**3.1. Functional requirements**

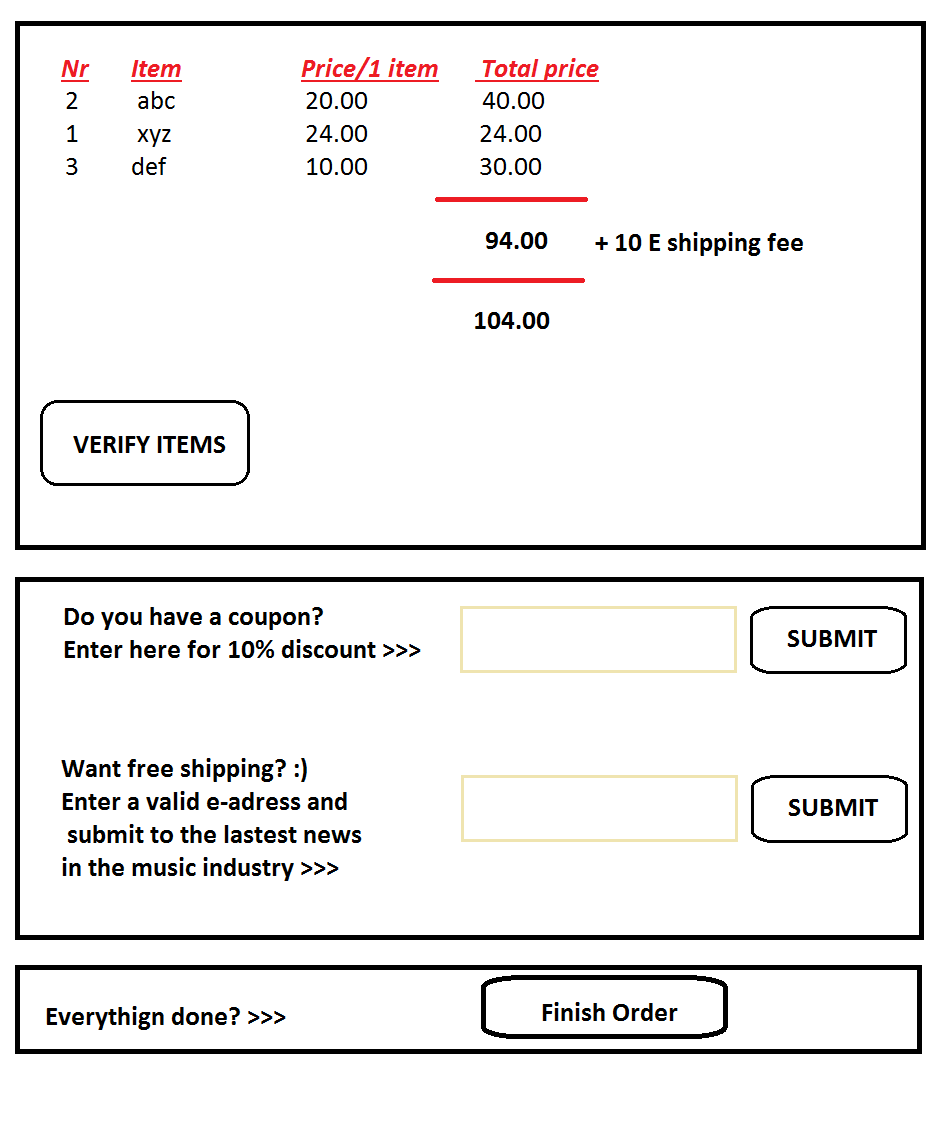
***× MAIN UI ×***

|  |  |  |
| --- | --- | --- |
| **ID** | **Affected entity** | **Description** |
| FR1 | The user | Should directly access the Main screen = the User Interface |
| FR2 | The user | Should have all the buttons available on the screen |
| FR3 | The user | Should see the list of available items = the full name and price |
| FR3 | The user | Can search an item by any part of its title by clicking the “Search” button |
| FR4 | The user | Can directly scroll through the list of items and click on a specific one to select it |
| FR5 | The user | Can add more of the same item to the cart |
| FR6 | The user | Can empty the items from the cart by clicking the “Reset” button |
| FR7 | The user | Can go to the checkout page by clicking on the “Checkout” button |

****

***× CHECKOUT UI ×***

|  |  |  |
| --- | --- | --- |
| **ID** | **Affected entity** | **Description** |
| FR8 | The user | Should access the Checkout UI after clicking on the “Checkout” button |
| FR9 | The user | Should see the list of items selected previously |
| FR10 | The user | Should see number of desired items, the title, price per 1 item and total price ( total price = nr of the same item \* price per 1 item) |
| FR11 | The user | Should see the SubTotal (price for the items alone) |
| FR12 | The user | Should see the Grand Total = SubTotal + 10 € shipping fee |
| FR13 | The user | Can verify if any item cannot be bought (see rules) |
| FR14 | The user | Can enter a coupon for a discount from the Grand Total  The coupon is “GMC10” for this case |
| FR15 | The user | Can enter an email address for free shipping |
| FR16 | The user | Can end the application by clicking the “Finish order” button |

****

## **3.2. Performance requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Affected entity** | **Description** |
| PR1 | Amount of money to be payed | Should be calculated precisely and accordingly to the user’s interaction |
| PR2 | Reset button | Should clear all the items already selected. |
| PR3 | Search button | Should display items related to the search criteria, not something else. |
| PR4 | Verify button | Should check the Drool rules accordingly |

## 

## **3.3. Interface requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Affected entity** | **Description** |
| IR1 | The user | The display should be big enough |
| IR2 | The user | The menus should have a good choice of color |
| IR3 | The user | Interface easy to use in general. |

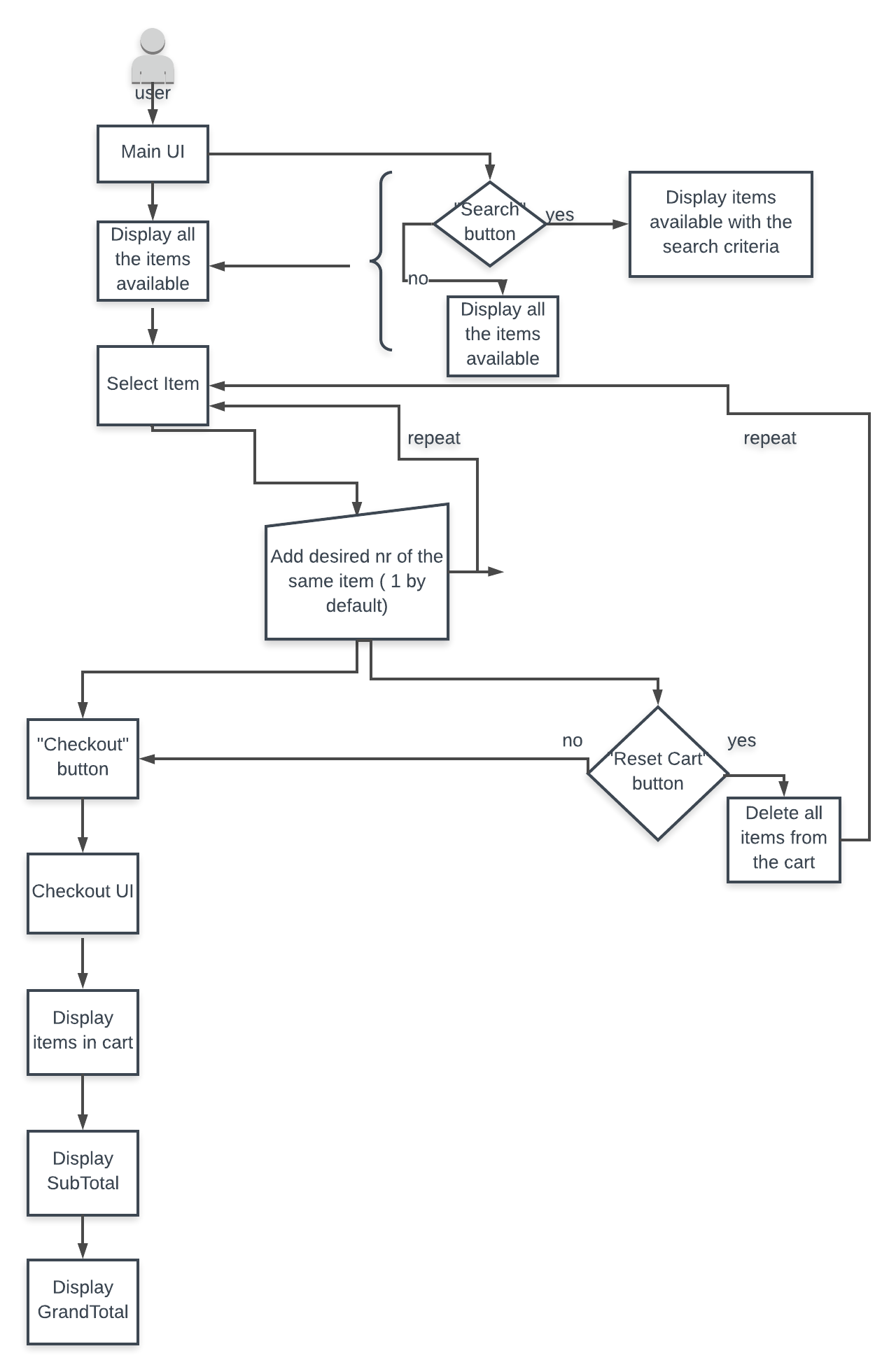
## 

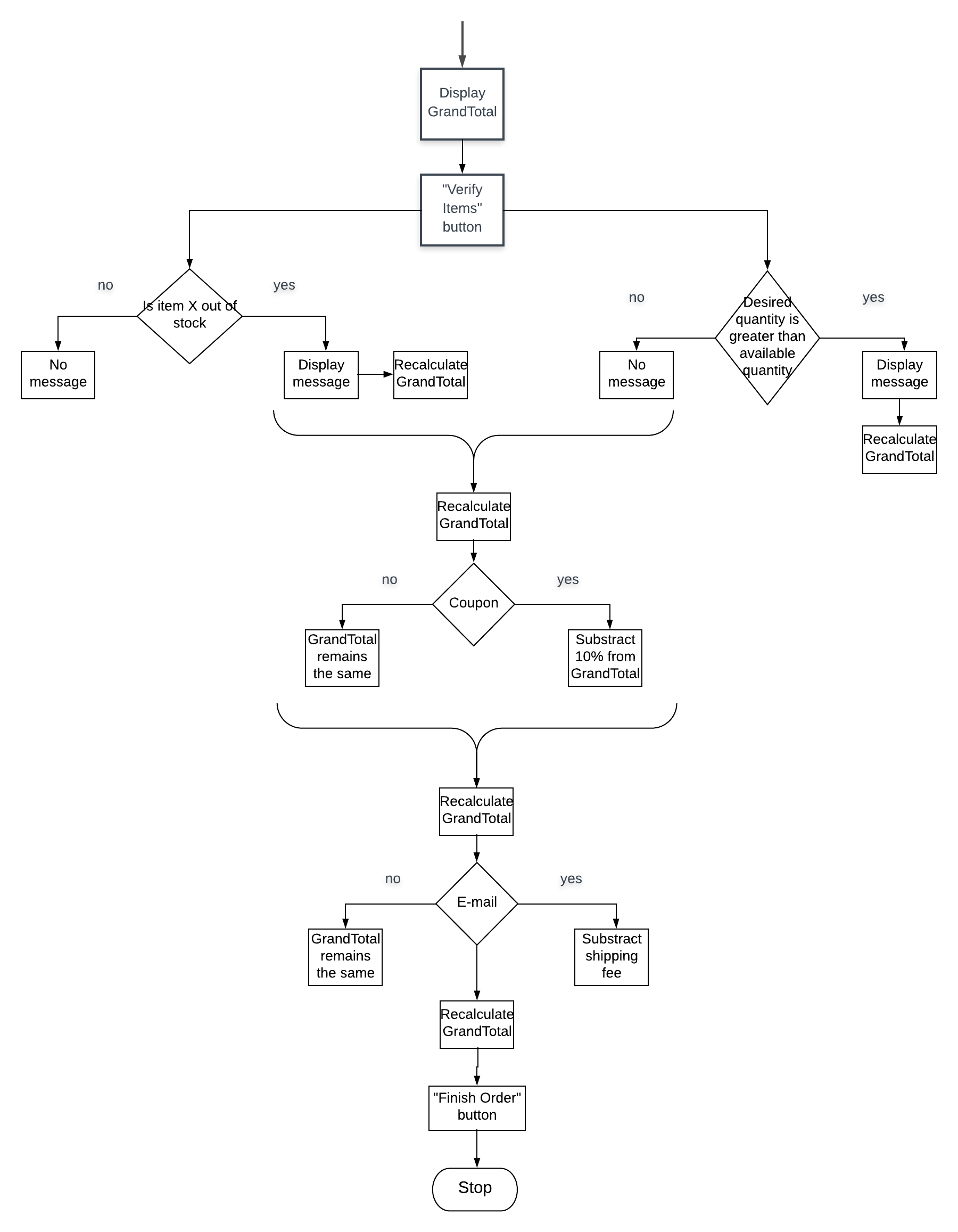
## **3.4. Software system attributes**

The application is as reliable, as it does not rely on Internet connection.

The application is portable, as it can be run on any device as long as the constrains paragraph 2.4 are respected.

**3.5. Use Cases Diagrams**



( continuation of the previous schema ) 

# 4. Non-functional requirements

* 1. Security
* There are no long-in requirements
* There are no security issues
  1. Performance
* System should not have a long response time
* System should not block or interrupt the usual performance of the user’s device
  1. Portability
* The software must be written character set-neutral
* The system will run on any Java target platform
  1. Availability
* The project is to due Monday, the 7th of January 2019
  1. Reliability
* The acceptance threshold for down-time is 1 minute
* The system will be available 100% of the time
* The system can never crash
  1. Compatibility
* The system is compatible with Windows Operating System, Windows XP and up of 32/ 64 bits
* The system requires Java SE 6 to be installed on the user’s device
  1. Modifiability
* Files can be modified only by the development team