

# Requirement Specification

For the StoreBase stock inventory application

# Table of Contents

<b>Introduction</b>	<b>4</b>
Logfile reporting	4
<b>Functional Requirements</b>	<b>5</b>
FR01: Adding a new product to the Stock Inventory	5
<i>F01-01</i>	5
<i>F01-02</i>	5
<i>F01-03</i>	5
<i>F01-04</i>	5
<i>F01-05</i>	5
FR02: Updating the Stock Inventory after Cargo	5
<i>FR02-01</i>	5
<i>FR02-01-01</i>	5
<i>FR02-01-02</i>	5
<i>FR02-02</i>	6
<i>FR02-02-01</i>	6
<i>FR02-03</i>	6
FR03: Updating the Stock Inventory after Warehouse Sale	6
<i>FR03-01</i>	6
<i>FR03-01-01</i>	6
<i>FR03-01-02</i>	6
<i>FR03-02</i>	6
FR04: Initialising the application	6
<i>FR04-01</i>	6
<i>FR04-01-01</i>	6
<i>FR04-01-02</i>	7
<i>FR04-01-03</i>	7
<i>FR04-01-04</i>	7
<i>FR04-02</i>	7
FR05: Starting up the Application	7
<i>FR05-01</i>	7
<i>FR05-02</i>	7
<i>FR05-03</i>	7

<i>FR05-04</i>	7
<b>Non-Functional Requirements</b>	<b>8</b>
NR01: Requirements related to the application build, maintainability and portability	8
<i>NR01-01</i>	8
<i>NR01-02</i>	8
<i>NR01-03</i>	8
NR02: Reliability of the application	8
<i>NR02-01</i>	8
<i>NR02-01</i>	8
<b>Domain requirements</b>	<b>9</b>
<i>DR01</i>	9
<i>DR02</i>	9
<i>DR03</i>	9
<i>DR04</i>	9

# Introduction

StoreBase will be an application usable for inventory management of rolls containing raw shoe lace. All stock items are specified by the following credentials.

Credential	Elucidation
artikelnummer	Reference code to the kind of lace
benaming	Name of the kind of lace
kleurcode	Reference code of the color of the lace
kleur	Name of the color of the lace
aantal meters	Length in meters of the lace on a roll
aantal rollen	Amount of rolls in stock
aantal op rol	
datum	Date of addition of a certain quantity to the inventory stock
prijs per meter	Price of lace in euro's per meter
totaal prijs	Total worth of lace in stock in euro's
totaal voorraad	Total amount of lace in stock in meters
QR code	Unique identity Quick Response code containing 'artikelnummer', 'kleur code', 'kleur', and 'datum'.

## Logfile reporting

As will be described in the requirements specification, after each action a line will be written to a log file. The list below shows the meaning of different report codes that are used for logging.

ReportCode	Condition
CredentialMapping	When a new 'artikelnummer'-'benaming' or 'kleur code'-'kleur' combination is added to the mapping list.
WriteSticker	When the credentials of a product item are written to the sticker file.
SoldItem	When the stock inventory is decreased after warehouse sale.

# Functional Requirements

Below, various functional requirements of the software are listed. These functional requirements describe what the software application should be able to work properly.

## FR01: Adding a new product to the Stock Inventory

The requirements below relate to the action of adding a new product. This occurs for example when a roll of lace is delivered in a color that has never been there before.

### F01-01

The user should be able to add a new product by entering the credentials after which the product is added to the stock database.

### F01-02

When the credentials of the product are added, the default value for 'datum' should be today. This value should be adjustable for usability. E.g. if the user wants to add products that were delivered earlier.

### F01-03

When the credentials of the product are entered, a unique QR code should be made by the application. This QR code should NOT be adjustable and be composed in the following way: <artikelnummer>'<kleur code>'<kleur>'<datum>.

### F01-04

If the new product contains unknown 'artikelnummer' and 'benaming', or unknown 'kleur code' or 'kleur', the combination should be mapped to the credential-mapping file as specified in **DR01**. This should be reported in the logfile with report code CredentialMapping.

### F01-05

The window for adding a new product should be opened from the main window by pressing a button [Add Product Item]

## FR02: Updating the Stock Inventory after Cargo

The following requirements relate to the action of updating the stock inventory after new products have been delivered. E.g. when four rolls of a certain lace are delivered, the database needs to be updated.

### FR02-01

After cargo, when products have arrived to the warehouse, the user should be able to enter the product credentials 'artikelnummer', 'kleurcode' and 'aantal rollen' after which the rest of the credentials should be filled automatically.

#### FR02-01-01

If the user scans a QR code instead of entering above recalled credentials, this should also work. However, the credential 'aantal rollen' should be filled manually.

#### FR02-01-02

If the user fills unknown 'artikelnummer' or 'kleur code', where unknown means it does not exist in the credential mapping file, the user should get a warning and must enter the corresponding 'benaming' and 'kleur'. These values should be mapped to the credential-mapping file.

## **FR02-02**

If the stock inventory of a product item is increased, the credentials 'artikelnummer', 'kleurcode', 'benaming', 'kleur', 'aantal op rol', 'datum', 'aantal rollen', and 'QR code' should be written on a sticker file. This action should be logged with ReportCode 'WriteSticker'.

## **FR02-02-01**

Each time the application is started, a new sticker file is made with filename equal to date-time and placed in the directory /stickers.

## **FR02-03**

The window for updating stock inventory after cargo delivery should be accessible from the main window by pressing the button [Update Stock Inventory].

# **FR03: Updating the Stock Inventory after Warehouse Sale**

The requirements below relate to the decrease of stock inventory. If a product item is taken from the warehouse, e.g. because the roll is empty due to use, the available amount of rolls in the warehouse should be updated in the database.

## **FR03-01**

The user should be able to scan the unique QR code of a product item when it's removed from the warehouse, after which the value of 'aantal rollen' is decreased (-1). This should be logged with ReportCode 'SoldItem'.

## **FR03-01-01**

If the stock inventory after warehouse sale is 0, the user should get a warning containing the product item credentials.

## **FR03-01-02**

If the stock inventory is 0 already before warehouse sale, the user should get a warning and should enter the stock inventory ('aantal rollen') to the current amount in stock. The user has to take the date of the product into account. Equal products that were added at different dates have different unique QR codes.

## **FR03-02**

When the stock inventory ('aantal rollen') is changed due to warehouse sale, the values of 'totaal voorraad' and 'totaal prijs' need to be recalculated. The database file must be updated after this in order to save changes.

# **FR04: Initialising the application**

The requirements below relate to the initialisation of the application and should happen before the main window pops up. The actions are necessary for the program in order to work properly.

## **FR04-01**

When the application is started, before the main window pops up, it should be made sure that the following recalled files and directories exist. If they don't, they should be made instantly. In that case, it means that there new ones made are empty.

## **FR04-01-01**

If it is not there, a directory /.bin should be created. In that directory, the files database.csv and credentialMapping.csv should be created. See **FR04-01-02** and **FR04-01-03** for file content.

### **FR04-01-02**

If the file database.csv is not in the directory /.bin it should be created instantly and have the following column fields: |'artikelnummer'|'benaming'|'kleurcode'|'kleur'|'aantal meters'|'aantal rollen'|'aantal op rol'|'datum'|'prijs per meter'|'totaal prijs'|'totaal voorraad'|'QR code'|.

### **FR04-01-03**

If the file credentialsMapping.csv is not in the directory /.bin, it should be created instantly and have the following column fields: |'code'|'name'|.

### **FR04-01-04**

In case it is not there, an empty directory /stickers should be created instantly.

### **FR04-02**

If the system starts up for the first time after going into a new year, all product items of which the amount in stock ('totaal voorraad') is 0 must be listed into a unique list.

## **FR05: Starting up the Application**

The requirements below relate to the actions that need to be performed when the application is started up, after the initialisation stage.

### **FR05-01**

While starting the application, a clean logfile should be made named after the current date-time. It should be stored in the .bin directory of the application.

### **FR05-02**

When the application is started, a main window should rise. From this window, the user should be allowed to access the different modes of the application: adding a new product, mapping credentials, and updating stock inventory. Each mode has its own window, and opening/closing one of these windows should be reported in the logfile.

### **FR05-03**

When the system is started up, a date-time value should be added to a general log file. This is necessary to keep track of the use of the application.

### **FR05-04**

When the application is started, the main window should show the last login, which it obtains from the general log file.

# Non-Functional Requirements

Listed below, the non-functional requirements are shown. These requirements relate to reliability, efficiency, usability, maintainability and portability.

## NR01: Requirements related to the application build, maintainability and portability

The following requirements relate to the build, maintainability, and portability of the application.

### NR01-01

The application should be written in Python and must support the most recent versions of the modules 'tkinter', 'csv', 'datetime', 'os', and 'numpy'. Execution of the program should be able with at least python version 3.8.7, given that the above recalled dependencies are installed.

### NR01-02

The application should be built using the most recent version of the py2app module. This module is not required for the functionality of the program but is fundamental for making the code into an executable, stand-alone application.

### NR01-03

The application should run on macOS 11 and higher, possibly extended to iOS and/or windows environments in later stages.

## NR02: Reliability of the application

The requirements below relate to reliability of the application. I.e. what does it take to make sure that data is kept safely and performed actions are logged.

### NR02-01

When changes are made to the database, i.e. if a new product is added or the stock inventory is updated, the updated values have to be written to the file and the file needs to be closed. Besides those actions, the database file should not be opened but the data should be loaded into the computer memory.

### NR02-01

Each time an action is performed, this should be logged into a log file that is created during start-up. This is applicable for any action besides starting up, and shutting down.



## Domain requirements

Domain requirements are requirements to the environment in which the software will be executed. Since StoreBase is an application specifically for stock inventory management of shoe laces, it's domain is quite narrow. Although there are a few requirements.

### DR01

The application has access to a file that matches the following credentials, as given in \ref{table1}, one-to-one.

Mapping of credentials	
artikelnummer	benaming
kleurcode	Kleur

### DR02

The application has access to a file holding stock inventory information for all used credential combinations. I.e. current inventory.

### DR03

The application has access to a logfile that contains a list of the changes made to the stock inventory since the application is started.

### DR04

The application has access to a general log file that stores the history of application usage.