PoS Systems

For <Subsystem or Feature>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20.10.16 | 1.0 | Creation of Document | Sven Baumann, Dominik Schneider, Sandra Kramlich |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Overall Description 4

3. Specific Requirements 5

3.1 Functionality 5

3.1.1 <Functional Requirement One> 5

3.2 Usability 5

3.2.1 <Usability Requirement One> 5

3.3 Reliability 5

3.3.1 <Reliability Requirement One> 6

3.4 Performance 6

3.4.1 <Performance Requirement One> 6

3.5 Supportability 6

3.5.1 <Supportability Requirement One> 6

3.6 Design Constraints 6

3.6.1 <Design Constraint One> 6

3.7 On-line User Documentation and Help System Requirements 6

3.8 Purchased Components 6

3.9 Interfaces 6

3.9.1 User Interfaces 7

3.9.2 Hardware Interfaces 7

3.9.3 Software Interfaces 7

3.9.4 Communications Interfaces 7

3.10 Licensing Requirements 7

3.11 Legal, Copyright, and Other Notices 7

3.12 Applicable Standards 7

4. Supporting Information 7

# Introduction

## Purpose

This SRS will give an overview for the requirements of this project. Our application is a product group based cash for retailers, connected to a small management system. It should be user-friendly, fulfill all legal requirements and offer all functions needed in a small retail.

## Scope

This software specification applies to the whole project and implies the cash and the management system.

## Definitions, Acronyms, and Abbreviations

SRS Software Requirement Specification

UCD Use-Case-Diagram

POS Point-of-sale

## References

## Overview

The following chapters are about our vision and the software requirements.

# Overall Description

Our ambition is to create an application, which fulfills all legal requirements and still is user-friendly. It should deliver all functions needed in a small retail. Therefore the user needs the possibility to create and manage product groups and employee numbers, which are used as identifier in the system. There should also be the possibility to create some statistics  with the old selling processes. During a selling process the user can select a product group and the price. After all products are added the user selects the payment methods and finishes the sales process. Different roles decide the access rights for employees. Depending on the rights, a user will be directly redirected to the cash or can decide rather he wants to use the cash or the management system.

# Specific Requirements

## Functionality

### Selection between Cash- and Management System Area

The user has the possibility to choose between the Cash-Area for selling processes and the Management- Area to manage the master data, depending on user rights.

### Management System

#### Employee Management

Manage employee data like cashier numbers, rights and passwords.

#### Costumer Management

Manage costumer date like name, surname, address, e-mail, etc.

#### Product Group Management

Manage product group date, like number, name and profit margin.

#### Statistics

Shows different statistics in a specific period of time.

##### Sales

Shows all sold articles with date, receipt number, cashier number, price of sale, sales discount and transaction volume.

##### Product Groups

Shows transaction volume separated by product group.

##### Cashier

##### Shows transaction volume separated by cashier.

### Cash

#### Cash Opening Entry

After starting the cash, the user has to enter the actual money paid in.

#### Enter Customer Number

If a costumer already has a costumer ID, the cashier can enter it for statistic purpose.

#### Add New Costumer

If a costumer hasn’t a costumer ID, the cashier can assign a new one.

#### Enter Product Group

The cashier can choose between multiple product groups. After selecting one, he has to enter the price and the entry is shown in a shopping basket list.

#### Sales Discount

Every entry in the shopping basket list can be selected to give sales discount.

#### Finish Sales Process

After every article is entered, the cashier can finish the sales process by asking the costumer to pay and printing the receipt.

#### Cash Check

The cashier is shown the sum of the transaction volume since the last cash opening.

#### Log out

A cashier can log out, to make way for another cashier.

#### Closing the Cash Accounts

At the end of a day of sales, the cashier has to close the cash accounts.

## Usability

### Instruction

The application should be as clearly and as easy to use as possible and shouldn't allow a wrong handling. Therefore the instruction time should be below 15 minutes.

## Reliability

### Availability

The application should be available 99% of time that retailers aren't restricted to a specific time, but the main utilisation time will be between 7am and 9pm.

### Mean Time Between Failures

The mean time between failures need to be at least one month.

### Mean Time to Repair

The mean time to repair depends on the failure. Small failures need to be fixed in about 15 minutes, bigger ones, including corrupt data in the database, should be done within a full hour.

### Accuracy

Since our application is handling financial data, there need to be a 100% accuracy.

### Bugs

There must not be any critical bugs, which lead to complete loss of data, corrupt data or wrong financial data. These need to be found during the tests. Significant bugs, like an inability to use certain parts of the application, also need to be prevented through extensive tests.

## Performance

### Response Time

To prevent slowing down the selling process, the transactions need to be completed in less than a second.

### Capacity

At the start there should be only one running system of the application, but it should be scalable for future use.

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.]

### <Supportability Requirement One>

[The requirement description goes here.]

## Design Constraints

[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

### Software Languages

- Java EE 8

- HTML

- CSS

### Architecture

Spring MVC framework.

## On-line User Documentation and Help System Requirements

The whole application will be built with an intuitive design, so there shouldn't be need for help, but an instruction with relevant processes with follow.

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so forth.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notices, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

# Supporting Information

[The supporting information makes the **SRS** easier to use. It includes:

Table of contents

Index

Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements.]