

**Software Requirements**

**Specification**

**for**

**Online shop**

**Beekeeping**

**Version 1.0**

**Prepared by Cristian Gabriel Spinu**

**Prepared for university project**

**24.03.2017**

# Table of Contents

[Table of Contents 1](#_Toc477090253)

[Revision History 1](#_Toc477090254)

[1. Introduction 2](#_Toc477090255)

[1.1 Purpose 2](#_Toc477090256)

[1.2 Product Scope 2](#_Toc477090257)

[1.3 Intended Audience and Reading Suggestions 2](#_Toc477090258)

[1.4 Definitions, Acronyms, and Abbreviations 2](#_Toc477090259)

[1.5 References 2](#_Toc477090260)

[2. Overall Description 3](#_Toc477090261)

[2.1 Product Perspective 3](#_Toc477090262)

[2.2 Product Functions 3](#_Toc477090263)

[2.3 User Classes and Characteristics 3](#_Toc477090264)

[2.4 Operating Environment 3](#_Toc477090265)

[2.5 Design and Implementation Constraints 3](#_Toc477090266)

[2.6 User Documentation 3](#_Toc477090267)

[2.7 Assumptions and Dependencies 4](#_Toc477090268)

[3. External Interface Requirements 4](#_Toc477090269)

[3.1 User Interfaces 4](#_Toc477090270)

[3.2 Hardware Interfaces 4](#_Toc477090271)

[3.3 Software Interfaces 4](#_Toc477090272)

[3.4 Communications Interfaces 4](#_Toc477090273)

[4. System Features 5](#_Toc477090274)

[4.1 System Feature 1 5](#_Toc477090275)

[4.2 System Feature 2 (and so on) 5](#_Toc477090276)

[5. Other Nonfunctional Requirements 5](#_Toc477090277)

[5.1 Performance Requirements 5](#_Toc477090278)

[5.2 Safety Requirements 6](#_Toc477090279)

[5.3 Security Requirements 6](#_Toc477090280)

[5.4 Software Quality Attributes 6](#_Toc477090281)

[5.5 Business Rules 6](#_Toc477090282)

[6. Other Requirements 6](#_Toc477090283)

[Appendix A: Glossary 6](#_Toc477090284)

[Appendix B: Analysis Models 6](#_Toc477090285)

[Appendix C: To Be Determined List 7](#_Toc477090286)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Comments** | **Version** |
|  |  |  |  |
|  |  |  |  |
| Cristian Gabriel Spinu | 24.03.2017 | First Revision | 1 |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of the Online Beekeeping shop application. It will explain the features of the system, the interfaces of the system and what the system will do.

## 1.2 Product Scope

The main purpose of the system is to enable customers to browse and order from any part of the world and hence increasing business scope.

## 1.3 Intended Audience and Reading Suggestions

## 1.4 Definitions, Acronyms, and Abbreviations

The **Shopping Cart** is a virtual store on the Internet where customers can browse the catalog and select products of interest

## 1.5 References

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# 2. Overall Description

## 2.1 Product Perspective

## 2.2 Product Functions

The product will have the following main functions:

**User: View Categories and Items:**

• 1: The users shall be able to view the categories on the application’s home page.

• 2: The users shall be able to view items in different categories.

• 3: The users shall be able add items to the cart.

• 4: The users shall be able to view more information about an item before adding it to the cart.

• 5: The users shall be able to able view the shopping cart.

• 6: The users shall be able to browse through the available items.

**User: View Shopping Cart:**

• 7: The users shall be able to view the items added to the cart.

• 8: The users shall be able to check out with the current items in the cart.

• 9: The users shall be able to continue shopping.

• 10: The users shall be able to delete items from the cart.

**User: Checkout Items**

• 11: The users shall be able to check out items only when there are items in the shopping cart.

Login/ User Authentication

• 12: The users shall login or register using the user authentication form.

• 13: The users shall not login or register if the information is incomplete or invalid.

**User: Place Order**

• 14: The users shall place an order by completing the information in the order form.

• 15: The users shall not be able to place an order if the information in the order form is invalid or incomplete.

**Admin: View User Information**

• 16: The administrator shall be able to view all the users’ information that completes the order form and the checkout process.

**Admin: Add/Update/Delete Shopping Items**

• 17: The administrator shall be able to add new items to the list of shopping items.

• 18: The administrator shall be able to modify/update an item’s price and description.

• 19: The administrator shall be able to delete items from the main page of the shopping-cart application.

**Additional Functional Requirements**

• 20: The administrator shall be able to view the entire history of the checked-out items.

• 21: The administrator shall be able to view the entire history for the users who successfully complete the checkout process.

## 2.3 User Classes and Characteristics

## 2.4 Operating Environment

The application supports the following web browsers Google Chrome and Mozilla Firefox.

## 2.5 Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## 2.6 User Documentation

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## 2.7 Assumptions and Dependencies

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# 3. External Interface Requirements

## 3.1 User Interfaces

The two interface types found in the online shopping-cart application are as follows:

1. User Interface: Users are able to view the home page of the shopping-cart application, browse the different categories, browse and add any number of items from any categories in the shopping cart, look for information about each product, delete the items in the shopping cart, save the cart for later viewing, check out or continue shopping after adding the item to the cart, and check out the items by completing the required information in the order form.
2. Admin Interface: The administrator is able to view the users’ information that was entered during checkout in the database, can update the information, price, shipping costs of the items, add or remove items from the main display.

## 3.2 Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## 3.3 Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## 3.4 Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# 4. System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## 4.1 System Feature 1

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

1. Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

1. Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

1. Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary.*

*Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:

REQ-2:

## 4.2 System Feature 2 (and so on)

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## 5.2 Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## 5.3 Security Requirements

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## 5.4 Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## 5.5 Business Rules

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# 6. Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

# Appendix A: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

# Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, ll,state-transition diagrams, or entity-relationship diagrams*.>

# Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*