**Documentation**

**for**

**ECommerce**

**Prepared by**

**Group Name:**

|  |  |  |
| --- | --- | --- |
| **Bacioiu** | **Alexandru Georgian** | **Section CEN 3** |
| **Balosu** | **Mihai Costinel** | **Section CEN 3** |
| **Popescu** | **Radu Andrei** | **Section CEN 3** |
| **Toma** | **Gigi Alexandru** | **Section CEN 3** |

|  |  |
| --- | --- |
| **Instructor:** | **C. Staoica Spahiu** |
| **Course:** | **Project II – Use of Databases** |
| **Date:** |  |

**Table of Contents**

**Table of Contents ii**

**1** **Introduction 1**

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Definitions, Acronyms and Abbreviations 1

1.4 References and Acknowledgments 1

**2** **Overall Description 2**

2.1 Product Functionality 2

2.2 End Users and Characteristics 2

2.3 System Stakeholders 2

2.4 Operating Environment 2

**3** **Specific Requirements 3**

3.1 User Interface 3

3.2 Functional Requirements 3

3.3 Use Case Diagram 3

**4** **Non-functional Requirements 4**

4.1 Software Quality Attributes 4

* **Introduction**
* **Document Purpose**

The purpose of this document is to provide the detailed description of the functionality of the TBPB web application. Within this documentation the user will find out the essential features of the application and how to take advantage of all our application services. This SRS also serves as an input for the designing and modeling of our shop. This document also covers hardware, software and various others technical dependencies.

* **Product Scope**

The project we want to implement is an online store. Within this application, we want to make a friendly environment for the customers, where they can easily access any of the application features.

There will be more roles, both normal customers(clients) and administrators. Each client has the opportunity to register and log in. The log in can be made either using the existing shop account or the Facebook account.

There is also a section dedicated to administrators, who can add products, including all the attributes that we will set. There are several product categories.

When logged in, each customer will be entitled to a shopping cart where they can add any of the existing products in the store. He can perform, in the shopping cart, operations to increase or decrease the quantity, but also to delete that product from the list. The price and quantities will be updated, with the addition or deletion of a product from the shopping list.

All accounts, all products and all shopping carts, along with the products they contain, will be stored in the database.

Each operation will be automatically updated in the database.

* **Definitions, Acronyms and Abbreviations**

HTML - Hyper Text Markup Language

HTTP – Hyper Text Transfer Protocol

IIS - Internet Information Service

Authentication - The procedure (essentially approval) used by the approval authority in verifying that specification content is acceptable. Authentication does not imply acceptance or responsibility for the specified item to perform successfully.

Client – (1) A computer process that requests a service from another computer and accepts the server's

responses; (2) the individual computers in a network computing system

Database - A collection of related data stored in one or more computerized files in a manner that can

be accessed by users or computer programs via a database management system.

Database management system - An integrated set of computer programs that provide the capabilities

needed to establish, modify, make available, and maintain the integrity of a database.

Functional requirement: A statement of a piece of required functionality or a behavior that a

system will exhibit under specific conditions. These include inputs, outputs, calculations, external

interfaces, communications, and special management information needs. Functional requirements

are also called behavioral requirements because they address what the system does.

JavaScript - A programming language designed by Sun Microsystems, in conjunction with Netscape,

that can be integrated into standard HTML pages.

Client – (1) A computer process that requests a service from another computer and accepts the server's responses; (2) the individual computers in a network computing system

Database - A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system.

Database management system - An integrated set of computer programs that provide the capabilities needed to establish, modify, make available, and maintain the integrity of a database.

Functional requirement: A statement of a piece of required functionality or a behavior that a system will exhibit under specific conditions. These include inputs, outputs, calculations, external interfaces, communications, and special management information needs. Functional requirements are also called behavioral requirements because they address what the system does.

JavaScript - A programming language designed by Sun Microsystems, in conjunction with Netscape, that can be integrated into standard HTML pages.

SQL (pronounced "ess-que-el") stands for Structured Query Language. SQL is used to communicate with a database. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases. Use SSMS to deploy, monitor, and upgrade the data-tier components used by your applications, and build queries and scripts.

6

Performance requirement -- A system/software system requirement specifying a performance

characteristic that a system/software system or system/software component must possess; for example,

speed, accuracy, and frequency.

PHP - An HTML-embedded Web scripting language

Portability - (1) A term used to describe an object that can be easily moved, such as a portable

computer; (2) When referring to computer software, portability refers to how easy a software program

can be moved between computer Operating Systems.

Requirement -A statement of need for some aspect of a system, often elicited directly from a stakeholder

or captured from a source document

Server – A central computer (server) which provides services such as file storage, printing, and

communications in a network computing system

Software requirement – (1) A software capability needed by a user to solve a problem to achieve an

objective; (2) A software capability that must be met or possessed by a system or system

component to satisfy a contract, standard, specification, or other formally imposed document.

System - A composite of equipment, skills, and techniques capable of performing or supporting

an operational role or both. A complete system includes all equipment, related facilities, material,

software, services and personnel required for its operation and support to the degree that it can be

considered a self-sufficient item in its intended operational environment.

System Requirement - A condition or capability that must be met or possessed by a system or system

component to satisfy a condition or capability needed by a user to solve a problem.

Use cases - A task analysis technique often used in software engineering. For each module of a

system, common tasks are written up with the prerequisites for each task, the steps to take for the user

and the system, and the changes that will be true after the task is completed. Use cases are especially

useful for making sure that common tasks are supported by the system, that they are relatively

straightforward, and that the system architecture reflects the task structure.

User class - A group of users for a system who have similar characteristics and requirements for the

system.

User interface – A user interface is what you have to learn to operate a machine. For examples, the

graphical user interfaces (GUIs) -- windows, icons, and pop-up menus have become standard on

personal computers.

User requirements - address what the users need to do their jobs. These requirements are implementation

independent and are sometimes called "business requirements." Read about the important of user

requirements.

Performance requirement -- A system/software system requirement specifying a performance characteristic that a system/software system or system/software component must possess; for example, speed, accuracy, and frequency.

Requirement -A statement of need for some aspect of a system, often elicited directly from a stakeholder or captured from a source document

Server – A central computer (server) which provides services such as file storage, printing, and communications in a network computing system

Software requirement – (1) A software capability needed by a user to solve a problem to achieve an objective; (2) A software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.

System - A composite of equipment, skills, and techniques capable of performing or supporting an operational role or both. A complete system includes all equipment, related facilities, material, software, services and personnel required for its operation and support to the degree that it can be considered a self-sufficient item in its intended operational environment.

System Requirement - A condition or capability that must be met or possessed by a system or system component to satisfy a condition or capability needed by a user to solve a problem.

Use cases - A task analysis technique often used in software engineering. For each module of a system, common tasks are written up with the prerequisites for each task, the steps to take for the user and the system, and the changes that will be true after the task is completed. Use cases are especially useful for making sure that common tasks are supported by the system, that they are relatively straightforward, and that the system architecture reflects the task structure.

User requirements -These requirements are implementation independent and are sometimes called "business requirements." Read about the important of user requirements.

* **References and Acknowledgments**

Word Definition:

<https://www.techopedia.com/definition/>

Different resources:

<https://en.wikipedia.org/>

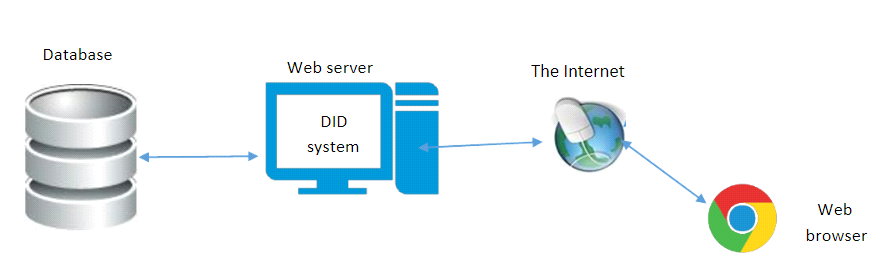
IEEE 830-1998 tamplate:

<http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>

Acknowledgments:

* **Overall Description**
* **Product Functionality**

The data system consists two main parts -- a central database management system for the uploading, storage, and management of data, and a client application to allow users access and interact with the data.



The product functions are more or less the same as described in the product perspective. The functions of the system include the system providing different type of services based on the type of users.

* There will be 2 types of users:
* Customers
* Administrators
* Each user can register and log in, using either the shop accounts, or the Facebook account.
* Users has the option to recover or change the password.
* Administrators will have a dedicated section, where they can manage the products.
* When logged in, each customer will have his own shopping cart, where they can perform actions, such as adding or deleting the quantity of a product. The total price will be also updated.
* There will be a limited number of products. When they are out of stock, it can not be added in the cart anymore.
* All the accounts, products and shopping carts will be stored in the database.
* Each operation will be updated in the database.
* **End Users and Characteristics**

The user class of this system will be:

* Maintenance and support team – engineers(developers) of the TBPB online shop, who are responsible for maintaining code and briefing system usability.
* User – any end user who wants to access the service provided by the TBPB shop
* System operator - TBPB information’s desk officer responsible for data archiving and maintenance, requiring full access for all aspect of the system i.e. read/write/modifying and has the privilege to manipulate all data in the system this include the role of commonly thought of as a Database administrator.
* **System Stakeholders**

For whom is the project usually created? A satisfactory answer will be — for end users. End users are also stakeholders on the project. However, they may not be the important ones. Therefore, in software development, it’s worth focusing not on end users, but entirely on stakeholders.

It’s impossible to compile a complete list of stakeholder types since, for different systems, they can differ significantly.

Let’s highlight the following stakeholders as the most common. Also, let’s look at each category in terms of consequences when ignoring their interests:

● Those who are involved in the project and work on it.

○ Project team. Imagine that you have developed a solution that uses the .NET technology stack. But there is one problem. You have twenty available Java developers in the company and no one who knows .NET. I suppose after you remember this, there will be no need in explaining why the designed solution is terrible. And this is the simplest example. You need to know the team to understand what technologies they know well and which of them should not be used just because they are trending.

○ Management team. Let’s say you forgot to ask your project manager about something important before making a decision. But they will cover you and do so for the designed solution to reach the final result. Moreover, often, a project manager has more information and listening to their opinion is very useful.

● Those who are affected by the project and who will use its artifacts, including the results.

○ Customers. Customers are one of the key stakeholders. If you are an architect, then there is only one question. How could you forget to discuss your decision with the people who pay the money for the project development? I will answer myself. It’s easy. In my practice, there was one example when a fantastic technical solution for real-time data processing and synchronization was created. This decision was one of the most advanced on the market, taking into account the latest technological trends. Furthermore, it was competently designed, correctly tested, and shown to the customer. And then it turned out that the customer wanted something different. More precisely, a completely different solution. And they did not need super synchronization at all.

○ End users. So, we finally got to them. I hope they always had a key influence on the project, but in practice, this is not true.

* **Operating Environment**

The rule for selecting hardware and software is that the components/application must be functionally efficient, capable of interfacing with other software, and easy to maintain.

* OE-1 - The System shall operate with the following Web browsers: Microsoft Internet Explorer, Safari, Chrome, Firefox and any other modern day browser.
* OE-2 – At least IntelCore I3 processor
* OE-3 – 4GB RAM
* OE-4 – Microsoft Windows 7 ++
* OE-5 – Internet Connection

# Specific Requirements

* 1. **User Interface**

1. Register page - Creates an account, the page has 3 input fields : e- mail, password and confirm password. In case you insert an e-mail address that's already used, a message will pop up

2. Log in page - Logs in the account, the page has 2 input fields : e- mail and password. Like the register page, in case of inserting an e-mail address or a password that does not appear in the database, a message will pop up.

3. Cart page - Here are displayed all the products that the online shop has. Every customer has a cart where can add or remove products.

4. Profile page - Edit your profile, for example : change your e-mail

address or password, add a phone number.

5. Add product page - Where you add a product and its price.

6. Home page - Users can see a describe about the shop.

## Functional Requirements

REQ 1: User registration is through email, password and confirm password.

REQ 2: For login, the user must enter the email and password

REQ 3: There must be a textbox where you enter your password.

REQ 4: If the password or the email does not coincide with those saved in the registration database, an error message will appear on the screen.

REQ 5: The message will be displayed in a new box.

REQ 6: There must be a button to send the information to a particular function.

REQ 7: There must be a function that receives information entered by the user, check the database and decide whether or not to grant further access

REQ 8: To click on the button we will have the following code:

protected void Button1\_Click(object sender, EventArgs e) {

Verify.user;

}

REQ 9: Upload products with their price and quantity.

REQ10: Displays information from the database will be presented to the user.

REQ11 : User can modify his information(email, password, phone number).

# Non-functional Requirements

## Software Quality Attributes

Write only one statement per line

Write only one declaration per line

Use implicit typing for local variables when the type of the variable is obvious from the right side of the assignment, or when the precise type is not important Use a try-catch statement for most exception handling

Use object initializers to simplify object creation

Call static members by using the class name

Use meaningful names for query variables

Use aliases to make sure that property names of anonymous types are correctly capitalized, using Pascal casing

Use implicit typing in the declaration of query variables and range variables

Use multiple from clauses instead of a join clause to access inner collections