

**Watch Web Shop**

Version 2.0

November 15, 2021

Prepared by: Berina Suljic

For: Authority Partners

Attn: Kenan Hadzirovic

Copyright © 2016 by Authority Partners.  
All rights reserved.

This document contains proprietary, trade secret and/or confidential information of Authority Partners. This document cannot be reproduced in any manner, electronic or otherwise, published or distributed in any form and by any means without advance written consent of Authority Partners.

The Authority Partners logo is a trademark of Authority Partners.

All other product and company names mentioned herein are the trademarks or service marks of their respective owners.

**Table of Contents**

[1. Purpose of this Document 5](#_Toc87862473)

[2.Executive Summary 5](#_Toc87862474)

[3. References 5](#_Toc87862475)

[4. Requirements 6](#_Toc87862476)

[4.1 User Stories 6](#_Toc87862477)

[4.1.1 User Story - SEARCH 6](#_Toc87862478)

[4.1.2 USER STOY – Products list 6](#_Toc87862479)

[4.1.3 USER STORY – PRODUCT DETAILS PAGE 6](#_Toc87862480)

[4.1.4 USER STORY – Buy product 7](#_Toc87862481)

[4.1.5 USER STORY – editing products 7](#_Toc87862482)

[4.1.6 USER STORY – PURCHASE HISTORY 7](#_Toc87862483)

[4.1.7 USER STORY – EMAIL NOTIFICATION 8](#_Toc87862484)

[4.2 Use Cases 8](#_Toc87862485)

[4.2.1 Use Case – SEARCH product 8](#_Toc87862486)

[4.2.2 Use Case – buy product 10](#_Toc87862487)

[4.2.3 Use Case – CHANGING PRICE 12](#_Toc87862488)

[5. Implementation solution 13](#_Toc87862489)

[5.1 Overview 13](#_Toc87862490)

[5.1.1 Design considirations 14](#_Toc87862491)

[5.1.2 Key concepts 14](#_Toc87862492)

[5.2 Architectural design 14](#_Toc87862493)

[5.2.1 Overview of modules / components 14](#_Toc87862494)

[5.2.2 Structure and relationships/dependencies 15](#_Toc87862495)

[5.2.3 Crosscutting Concerns 15](#_Toc87862496)

[5.3 Detailed design of components 15](#_Toc87862497)

[5.3.1 Data Design 15](#_Toc87862498)

[5.3.2 Activity diagrams 16](#_Toc87862499)

[5.4 User interface 18](#_Toc87862500)

[5.4.1 PRODUCTS LIST PAGE 18](#_Toc87862501)

[5.4.2 PRODUCT DETAIL PAGE 19](#_Toc87862502)

[5.4.3 SHOPPING CART PAGE 20](#_Toc87862503)

[5.4.4 PURCHASE HISTORY PAGE 21](#_Toc87862504)

[5.4.5 ADMINISTRATOR DASHBOARD 22](#_Toc87862505)

[5.4.6 ADD NEW WATCH FORM 23](#_Toc87862506)

[5.4.7 EDIT WATCH FORM 24](#_Toc87862507)

# 1. Purpose of this Document

This document provides details about functional or non-functional requirements for Watches Web Shop application. It is explained what this application works, how does it do it and how does users use this application. With everything written here about use cases, design, architecture and other details, this documentation helps the programmer to stay within the context and understand it thoroughly.

The main focuses of this document are user experience, development, and implementation solution. By writing this document, we avoid risks of missing out on key requirements or creating gaps in understanding what the system is expected to do. Every requirement is processed in detail, so programmer understand it much more. But this document is still not so extensive and covers only those details that are necessary to understand how system works and what it should provide. Everything needed to start the implementation is available here and very easy to understand even if this project is not done by the person who wrote this documentation.

It is a known fact that whether it is for a small or big project, documentation is a critical activity that can improve the quality of the final product that is delivered, so that is one of the reasons for writing this document. Requirement documentation provides relevant information about expected functionalities and the use of the proposed software. These requirements are based on project requirements described in Ap Web Shop Project Requirements document.

It can also help end users of the system to understand how the system works, although this will be clear because the developer will try to make the design of the user interface as good as possible.

This document may change during the development cycle, but most things will remain unchanged because the system is expected to work exactly as described here. And that is the main purpose of this document, to be guided by it during development

# 2.Executive Summary

At the beginning of this document, few user stories are adduced and detailed acceptance criterium for them. There are described basic user stories and some tricky situations, in addition to give some introduction into requirements. Also, after processing user stories, use cases for some user stories are displayed with use case diagrams and other details such as actors and conditions. Further, a detailed implementation solution is presented, key concepts, goals, design considerations, as well as the architectural design. This document also contains detailed design of components with ERD and Activity Diagrams. Finally, the design of the user interface is presented and there the wireframes are illustrated with detailed description.

# 3. References

Ap Web Shop Project Requirements, Rihad Seherac, Kenan Hadzirovic, October 29, 2021

# 4. Requirements

## 4.1 User Stories

### 4.1.1 User Story - SEARCH

As a Customer, I want to be able to search products in different ways so I can find the watch I want.

Acceptance Criteria:

* 1. Searchtextbox in the middle is present
  2. Categories dropdown has categories: gender, brand and style
  3. Select different filters to reduce the number of irrelevant items and display filter results
  4. Sort items by alphabet, price, price + shipping, estimated delivery time…ascending and descending and display it
  5. Navigate between pages, from homepage to product list works

### USER STOY – Products list

As a Customer, I want to be able to see products list so I can view all the products.

Acceptance Criteria:

1. Product list is displayed on homepage or whenever user searches for product with any method
2. Products list contains product card for each product
3. Product card has image, name, price of product and button for add to cart and buy now if user is logged in
4. Product list is paginated and showed as a grid

### USER STORY – PRODUCT DETAILS PAGE

As a Customer, I want to be able to see Product Details Page so I can find out more information about watch I selected.

Acceptance Criteria:

1. Customer can navigate to Product Details Page
2. Customer can click on some product to open the details about it and details display to Customer
3. Customer can see list of details about product (brand, model, date of publishing, styles, bracelet details, case details, gender, condition, color, price, shipping price, time of shipping, guarantee, and detailed description

* User can see image of product, rating, and user reviews

### USER STORY – Buy product

As a registered Customer, I want to add product to cart so I can buy a product.

Acceptance Criteria:

1. Customer is registered and can log in to
2. ‘Buy now’ button is present
3. ‘Add to cart’ button is present
4. ‘Check out’ button is present
5. Quantity of available products is displayed
6. Item price and shipping price for product are displayed
7. Customer’s cart is displayed by clicking on shopping cart in header
8. Customer’s subtotal and total price for product in cart is displayed

### USER STORY – editing products

As an administrator, I want to be able to Add, Modify and Delete products, so I can provide information and market to users.

Acceptance Criteria:

1. Log in as administrator works
2. Navigate to page for Add, Modify and Delete products works
3. Button for Add, Modify and Delete product in last column of table of products are present
4. Form for Add new product with fields to enter details of products is displayed
5. Form for Editing existing product with filled fields of details about that product is displayed
6. Additional confirmation within special dialog before deleting product
7. Have Cancel and Save buttons on forms

### USER STORY – PURCHASE HISTORY

As a registered user, I need to be able to see Purchase details page so I can review all my previous purchases.

Acceptance Criteria:

1. Log in as registered Customer
2. Navigate to Purchase details page where is displayed table of purchased items works
3. Date and time of the purchases is displayed
4. Total price of the purchases is displayed
5. Product deletion or changing products price do not affect information in purchase history

### USER STORY – EMAIL NOTIFICATION

As a registered user, I want to get email notification so I can get notified about recent cart activity.

Acceptance Criteria:

1. Each time when user adds some items into cart, or performs any updates to existing items in cart, delayed email notification is triggered
2. Notification takes place 15 minutes after user’s last activity on the site.
3. Notification will not be sent if cart is empty

## Use Cases

### 4.2.1 Use Case – SEARCH product

Diagram

Description automatically generated

#### 4.2.1.1 Summary

This Use Case shows the behavior of the System when Customer is searching for the product. A set of actions that are necessary to lead the customer to the result is shown. The driver of this whole use case is the user who interacts with System. There are different ways to search for the product, and they are displayed in this use case.

#### 4.2.1.2 Used By

User Story – Search uses this use case.

#### 4.2.1.3 Pre-Conditions

Customer doesn’t have to be registered to use this use case, so there are none pre-conditions.

#### 4.2.1.4 Success End-Conditions

After successful finish of this use case (buying products), system has no change.

#### 4.2.1.5 Failed End-Conditions

If use case has finished with failure, Customer didn’t find the product probably because the system has not the one user was looking for.

#### 4.2.1.7 Acceptance Criteria

Normal flow of this use case is:

* Customer enters the page successfully
* Customer searches for product in three ways

1. Customer uses search textbox in the middle of the page and enters some text there, pick a category he wants to search and clicks on button Search
2. Customer uses categories dropdown, which is displayed left to search box, and picks up a category he want to be displayed
3. Customer immediately lists product list to search

* Displaying results of search
* Customer uses filters to filter results or sort it

#### 4.2.1.8 Actors

Customer is only actor that interacts with system.

##### 4.2.1.8.1 Primary Actor

Customer is the primary actor.

##### 4.2.1.8.2 Other Actors

Database and other services are other actors with whom the system interacts to perform search and get the products to display.

#### 4.2.1.9 Trigger

The action upon the system that starts this use case is user that searches for product.

### 4.2.2 Use Case – buy product

Diagram

Description automatically generated

#### 4.2.2.1 Summary

This Use Case shows the behavior of the System when Customer is buying the product. A set of actions that are necessary to lead the customer to the final purchase of the product is shown. The driver of this whole use case is the registered Customer who interacts with System. This use case is the most important use case of this application because as the name says, it is a Web Shop, so the main functionality is buying products.

#### 4.2.2.2 Used By

User Story – Buy product uses this use case.

#### 4.2.2.3 Pre-Conditions

Customer has to be registered and logged in to system, so he has the ability to click on buttons Add to Cart or Buy Now because these buttons are displayed only to users who are logged in. System has to have some products so that the user can select the product.

#### 4.2.2.4 Success End-Conditions

After successful finish of this use case (buying products), system has fewer products in the database because some of them are sold.

#### 4.2.2.5 Failed End-Conditions

If use case has finished with failure, system has no changes because user didn’t buy any product, so there is no change.

#### 4.2.2.7 Acceptance Criteria

Normal flow of this use case is:

* Customer is logged in successfully
* Customer searches for product successfully by using search textbox or search by categories or just by reviewing the product list
* Checking out in three ways

1. Customer view product and there he Add to cart or Buy now
2. Customer Add to cart and then he can search for more items or immediately go to cart list Check out
3. Customer Buy now

* Show product checkout page
* Check out and finish the shopping

#### 4.2.2.8 Actors

Registered Customer is only actor that interacts with system.

##### 4.2.2.8.1 Primary Actor

Registered Customer is the primary actor.

##### 4.2.2.8.2 Other Actors

Database and other services are other actors with whom the system interacts to perform some functionalities.

#### 4.2.2.9 Trigger

The action upon the system that starts this use case is Customer that tries to buy product.

### 4.2.3 Use Case – CHANGING PRICE

Diagram

Description automatically generated

#### 4.2.2.1 Summary

This Use Case shows the behavior of the System when user logged in as administrator tries to change price of the product. A set of actions that are necessary to change the product is shown. The driver of this whole use case is the registered administrator who interacts with System. This use case has some tricky case for price of items in the cart and items in purchase history.

#### 4.2.2.2 Used By

User Story – Editing product uses this use case.

#### 4.2.2.3 Pre-Conditions

User has to be registered as administrator and logged in to system, so he has the privilege of editing the product.

#### 4.2.2.4 Success End-Conditions

After successful finish of this use case, system has different price for some item, and Customer who had that item in cart gets notification that the price has changed.

#### 4.2.2.5 Failed End-Conditions

If use case has finished with failure, system has no changes.

#### 4.2.2.7 Acceptance Criteria

Normal flow of this use case is:

* User is logged in successfully as administrator
* Administrator selects item for modify and gets navigated to modify page
* Administrator enters new value of price and saves it
* Price changes and Customer that has that product in cart get notified

#### 4.2.2.8 Actors

Registered administrator is only actor that interacts with system.

##### 4.2.2.8.1 Primary Actor

Registered administrator is the primary actor.

##### 4.2.2.8.2 Other Actors

Database and other services are other actors with whom the system interacts to perform some functionalities.

#### 4.2.2.9 Trigger

The action upon the system that starts this use case is administrator that tries to modify product.

# 5. Implementation solution

This section provides appropriately detailed implementation solution.

## 5.1 Overview

Introduction section gives short overview of solution concept, assumptions, and dependencies.

### 5.1.1 Design considirations

This application is going to be divided into layers such as UI, API, BLL, DAL, DTO, Model and all of them has their own job. UI layer will provide user interface. API layer is responsible for handling server requests. BLL layer will contain business logic and it handles services that call repositories in the DAL layer. The DAL layer communicates with the database and converts to DTO classes.

#### 5.1.1.1 Assumptions and dependencies

The software is dependent on the publishers. Publishers get the opportunity to market their products on our website, so we have a large product datasets and traffic on our page. We can assume that this software will be used by a lot of people, and we need to give the users the most intuitively and comfortable interface.

#### 5.1.1.2 General Constraints

This system is constrained by amount of watches it has in database because if there aren’t enough watches, user won’t find the watch he is looking for. We should provide a lot of choices to user, so our application would work as it should.

#### 5.1.1.3 Goals and Guidelines

The biggest goal of this application is to provide the best possible interface to user. User should use this system as easier as possible, and every functionality should work just like user sees it. These functionalities should have the best performance as possible.

### 5.1.2 Key concepts

The key concept is the division of the application which provides us well structured and organized project, so our software development is going to be better and easier to organize.

## 5.2 Architectural design

High level architectural design including the list of all relevant existing components in correlation with new components.

### 5.2.1 Overview of modules / components

The application will be organized into 3 tiers. First tier is a single page application running in browser. Second tier is a .NET Core application which has several different layers which separate server functionalities. Third tier is a MS SQL database.

The first application tier is going to contain Angular application which will provide us a great design of user interface.

In the second application tier we are going to have our API and all the business and database logic. This tier will be parted into more layers, so all the components are organized well.

The last tier represents database tier where all the data will be stored.

### 5.2.2 Structure and relationships/dependencies

Diagrams describing logical modules relationships and dependencies can be found in one of the chapters below.

### 5.2.3 Crosscutting Concerns

Rules and guidelines for crosscutting concerns such as instrumentation and logging, authentication, authorization, exception management, communication, and caching.

## 5.3 Detailed design of components

### 5.3.1 Data Design

There is presented Entity Relationship Diagram.

Graphical user interface, diagram

Description automatically generated

### 5.3.2 Activity diagrams

Diagram

Description automatically generatedThere is presented activity diagram for shopping and buying products.

## 5.4 User interface

### 5.4.1 PRODUCTS LIST PAGE

This wireframe shows design of Products List Page. It should contain a header on the top, filters on the left, sort by behind product cards, grid of product cards and it should implement pagination. Product card contains image of watch, name (shown as brand + model), rating and price. Additionally, when user hang over with mouse, borders of product card display and icons for adding product to Wishlist and Cart. This page doesn’t require some validation because user just clicks on controls on page. If search has no result, this page doesn’t have product cards, but just a supportive message to Customer like “No search results”.

Diagram

Description automatically generated

### 5.4.2 PRODUCT DETAIL PAGE

This wireframe shows what Product Detail Page should contain. As we see in the picture, it should contain header, product image on the left. On the right, it should display product details, such as Brand Name, Model Name, Rating, Reviews, Product Description, and other details. It must contain Price and Quantity which is user’s choice. Below the details, there are buttons Buy Now that leads to Check Out Page and Add to Cart that adds product to Cart with chosen quantity. This page also has icon for adding watch to Wishlist.

Graphical user interface

Description automatically generated with low confidence

### 5.4.3 SHOPPING CART PAGE

This wireframe shows layout for Shopping Cart Page. It should contain header, list of products in cart, button for Check Out and total price in cart. Every item in list has attributes: product details (image, brand + model), quantity, price and total price with shipping. There is no validation too, but when user clicks on Check Out, it should lead him to Check Out Page where he enters some information.

Graphical user interface, text, email

Description automatically generated

### 5.4.4 PURCHASE HISTORY PAGE

There are displayed header and list of products that user has bought. Every item in list contains attributes: product details, quantity, purchase price, total price, and time of purchase. It should implement pagination too.

Graphical user interface, text, email

Description automatically generated

### 5.4.5 ADMINISTRATOR DASHBOARD

Bellow is displayed wireframe for Administrator Dashboard Page. Administrator can Add new Watch by clicking on button Add new Watch. He can also review all products that are currently in database, and he can edit or delete them.

Table

Description automatically generated

### 5.4.6 ADD NEW WATCH FORM

This is form for adding new watch. Administrator should fill this information: Brand, Model, Date Published, pick Styles, Gender, Condition, Color, Bracelet Material, Case Material, Case Diameter, Water Resistant, Price, Shipping Price and Guarantee field. All this information should be validated. When administrator clicks Save, system should display a supportive message that product is added, if it is verified. If information isn’t filled in correct way or there is missing some information, there should appear an Alert with Error Message.

Graphical user interface

Description automatically generated

### 5.4.7 EDIT WATCH FORM

This form is exactly the same as Add New Watch Form, but this information is already filled, and administrator can edit it. Only Brand, Model and Date Published are disabled. System should validate and show messages to user just like with Add Form.

Graphical user interface

Description automatically generated