

**Software Requirements**

**Specification**

**For**

**Web Aplication Design for Antique Shop Online**

**Version 1.0**

**Prepared by**

**Nicolae Ovidiu Petrisor**

**19.03.2017**

# Table of Contents

[Table of Contents 1](#_Toc477389832)

[Revision History 1](#_Toc477389833)

[1. Introduction 2](#_Toc477389834)

[1.1 Purpose 2](#_Toc477389835)

[1.2 Product Scope 2](#_Toc477389836)

[1.3 Intended Audience and Reading Suggestions 2](#_Toc477389837)

[1.4 Definitions, Acronyms, and Abbreviations 2](#_Toc477389838)

[1.5 References 2](#_Toc477389839)

[2. Overall Description 3](#_Toc477389840)

[2.1 Product Perspective 3](#_Toc477389841)

[2.2 Product Functions 3](#_Toc477389842)

[2.3 User Classes and Characteristics 3](#_Toc477389843)

[2.4 Operating Environment 3](#_Toc477389844)

[2.5 Design and Implementation Constraints 3](#_Toc477389845)

[2.6 User Documentation 3](#_Toc477389846)

[2.7 Assumptions and Dependencies 4](#_Toc477389847)

[3. External Interface Requirements 4](#_Toc477389848)

[3.1 User Interfaces 4](#_Toc477389849)

[3.2 Hardware Interfaces 4](#_Toc477389850)

[3.3 Software Interfaces 4](#_Toc477389851)

[3.4 Communications Interfaces 4](#_Toc477389852)

[4. System Features 5](#_Toc477389853)

[4.1 Activity diagram 6](#_Toc477389854)

[4.2 Activity diagram 7](#_Toc477389855)

[5. Other Nonfunctional Requirements 8](#_Toc477389856)

[5.1 Performance Requirements 8](#_Toc477389857)

[5.2 Security Requirements 8](#_Toc477389858)

[5.3 Software Quality Attributes 8](#_Toc477389859)

[5.4 Business Rules 8](#_Toc477389860)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Comments** | **Version** |
|  |  |  |  |
|  |  |  |  |
| Nicolae Ovidiu Petrisor | 19.03.2017 | First Revision | 1 |

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a description of a antiques online shop . It will explain the features of the system, the interfaces of the system and what the system will do.

## 1.2 Product Scope

The system will be built on a client/server architecture for assisting online customers on achieving the following tasks:

* Be able to see the products.
* Be able to add them to register and login.
* Be able to add items on cart.
* Be able to buy items.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for both the customers and the developers of the system.

## 1.4 Definitions, Acronyms, and Abbreviations

Used terms manage cart, view antiques, make purcashe.

## 1.5 References

GitHub project link: https://github.com/Ovidiu-Petrisor-Nicolae/W.A.D.

# 2. Overall Description

## 2.1 Product Perspective

This product is an online shop. The purpoase is to be able to buy items by the customers.

## 2.2 Product Functions

The product will have the following main functions:

* Guest Customer are able to register, view the items and buy items.
* Registered User can do what Guest Customer do, plus he have a cart where he can add, delete or edit items.
* An Authentication is a <<Services>> where Admin are able to modify items and verify the register

## 2.3 User Classes and Characteristics

The application users will be the Mayor (full privileges) High level employees (full privileges), low level employees which will have limited privileges and citizens which will have regular accounts.

## 2.4 Operating Environment

This application was developed in:Microsoft Visual Studio IDE.

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

## 2.5 Design and Implementation Constraints

This application is distributed with a proprietary software license.

The owner of proprietary software exercises certain [exclusive rights](https://en.wikipedia.org/wiki/Exclusive_right) over the software. The owner can restrict use, inspection of source code, modification of source code, and redistribution.

## 2.6 User Documentation

System Requirements Specifications document, System Design Specifications document and testing documents.

## 2.7 Assumptions and Dependencies

An internet connection is required in order to use the application.

Introduced data must respect the formatting options of the field in case.

# 3. External Interface Requirements

## 3.1 User Interfaces

The application should be accessible via web.  
The Layout should be the same for all users, with a main page for announcements and general information, and a navigation bar that allows access to the other functionalities and pages depending on the type of user.

Regular userswill be able to:

* Establish an online (individual/group) meeting with a municipality employee.
* View the documents and announcements of the city hall.
* View the organizational chart.

Full privilege will be able to

* Can modify audience schedule.
* Can add/remove/modify documents
* Can create/delete announcements.
* Respond to user message requests.

## 3.2 Hardware Interfaces

The application will be supported by any desktop or laptop device.

## 3.3 Software Interfaces

The application is connected to the database (SQL) through a connection string located in the web.config file from the application project.

There will be a communication interface between the server and the database (functions to read from and write into the database) and another one between the server and the client application (get input text and display data).

## 3.4 Communications Interfaces

This application uses the HTTP protocol.

# 4. System Features

* **Home Page**

The home page consists of the main page of the Municipality.

It will have a main body that contains News/Announcements from the Municipality.

It will have a navigation bar at the top of the page that will have links to the other features of the site, and will display the current logged user, or links to Log In or Register in case of no logged user.

* **Log in**

Will consist of:

A text box for username

A textbox for password

A link for recovering the password

A log in button

* **Register**

Will consist of:

A text box for username

A textbox for password

A textbox for email

A register button.

* **Organizational chart**

Will consist of a graphical representation of the organizational chart or a table representation of each department, users with sufficient privileges will have options enabled to edit and modify such tables.

* **View Documents**

Will consists of a table of documents of interest, depending on user privilege, they will be shown all or only part of them, also depending on user privilege, updating, downloading, and approval options will be enabled or disabled for each entry depending on case

Each entry will have a log link that will be visible by users with proper privilege, such link shall redirect the user to a page with a list that contains all user interactions with said document.

An Upload Document button should be available for users with privilege to do so.

* **Online audience schedule**

Will consist of a form accessible to simple users that allows scheduling and requesting an audience.

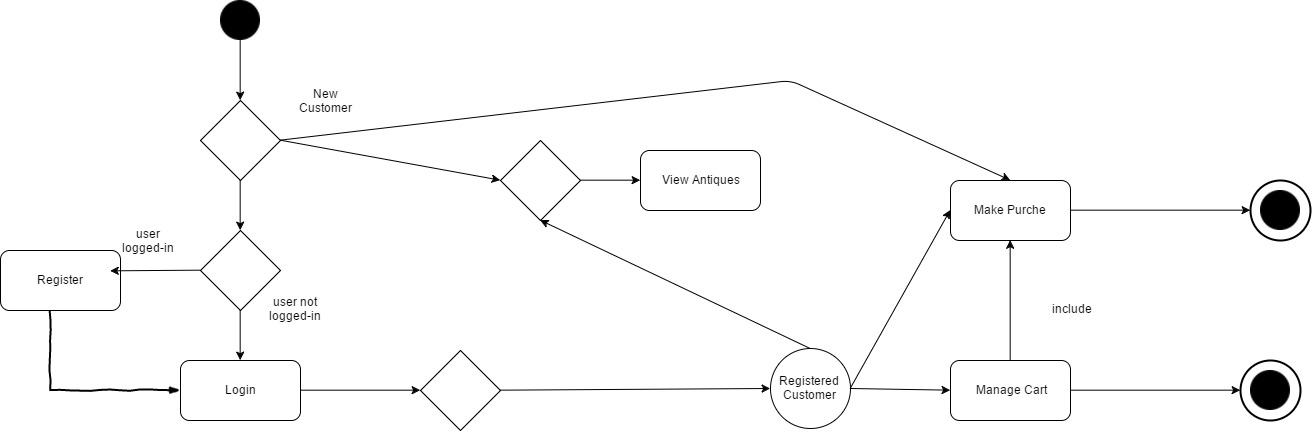
* **Manage audience schedules**

Visible only to High privilege users, it will consist of a table of all scheduled audiences with options to edit details for each entry.

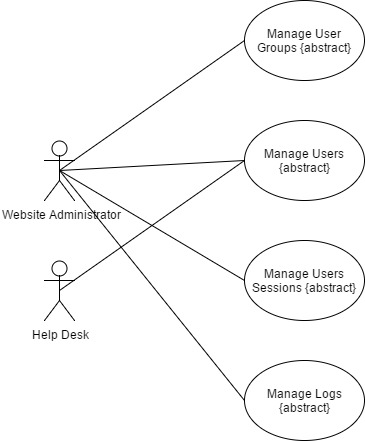
* **Audience Interface**

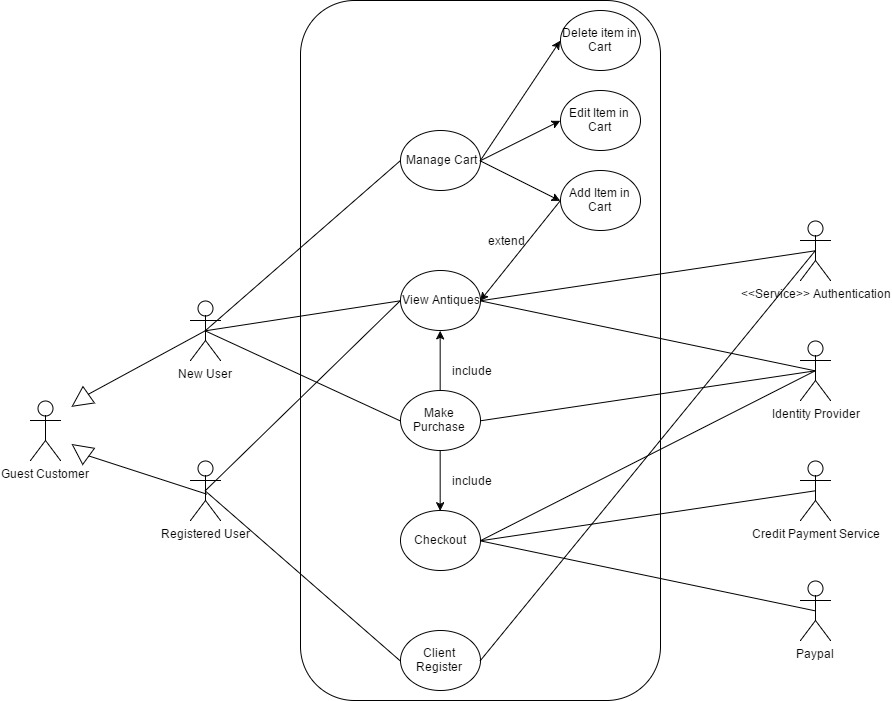
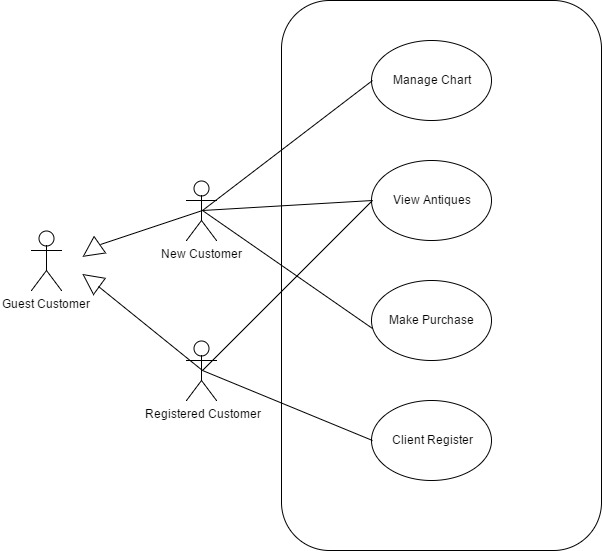
Will consist of a simple chat interface in which municipal employees and users can communicate in scheduled audiences.

## 4.1 Activity diagram



## 4.2 Use Case diagram





# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

1. The information system will run on Windows 7/8/10 OS.
2. Minimal system requirements:

* Processor: 3 GHz
* RAM Memory: 128 Mb
* Storage space: 10 Mb minimum

1. An internet connection should be set up before running the web application.

## 5.2 Security Requirements

The sensitive information in this application is considered:

* User and password – password will be encrypted using MD5 hash
* Database – Protection against SQL Injection will be set to every necessary input that works with the database

## 

## 5.3 Software Quality Attributes

Rule 1 (required): No reliance shall be placed on undefined or unspecified behavior.

Rule 2 (required): Insert one space between the comment delimiter (//) and the comment text

Rule 3 (advisory): No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure member and union member names.

Rule 4 (advisory): Write only one declaration per line.

Rule 5 (advisory): Add at least one blank line between method definitions and property definitions.

Rule 6 (advisory): Place the comment on a separate line, not at the end of a line of code.

Rule 7 (advisory): Use the + operator to concatenate short strings.

Rule 8 (advisory): Use meaningful names for query variables.

## 5.4 Business Rules

There is no warranty or representation, expressed or implied, as to the accuracy, completeness or appropriateness of the information on this website. Therefore the user must assume full responsibility for using the information and agree that developers are not responsible or liable for any claim, loss, damage or inconvenience caused as a result of reliance on such information.  
All rights reserved.  
Images and copy on this website are for personal use only and cannot be reproduced commercially without permission.