
Software Requirements Specification

Real estate shop

Prepared by Cristian Alexandrescu

Faculty of Automation, Computers and Electronics

Version 1.0

March 12, 2017

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope	1
1.4 References	1
2. Overall Description	1
2.1 Product Perspective	1
2.2 Product Functions	1
2.2.1 Reading offers	1
2.2.2 Login panel	2
2.2.3 Register panel	2
2.2.4 Add/Modify/Remove offer	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	4
3.3 Software Interfaces	4
3.4 Communications Interfaces	4
4. System Features	4
4.1 Minimum front-end development	4
4.1.1 Description and Priority	4
4.1.2 Stimulus/Response Sequences	4
4.1.3 Functional Requirements	4
4.2 Minimum back-end development	5
4.2.1 Description and Priority	5
4.2.2 Stimulus/Response Sequences	5
4.2.3 Functional Requirements	5
4.3 Improvements for the front-end interfaces	5
4.3.1 Description and Priority	5
4.3.2 Stimulus/Response Sequences	5
4.3.3 Functional Requirements	5
4.4 Late back-end development	6
4.4.1 Description and Priority	6
4.4.2 Stimulus/Response Sequences	6
4.4.3 Functional Requirements	6
5. Other Nonfunctional Requirements	6
5.1 Performance Requirements	6
5.2 Safety Requirements	6
5.3 Security Requirements	6
6. Other Requirements	7

Revision History

Name	Date	Reason For Changes	Version
Cristian Alexandrescu	03/12/2017	Initial Version	1.0

1. Introduction

1.1 Purpose

The purpose of this document is twofold:

1. To explain the process by which the preliminary requirements were analyzed and refined by the teacher
2. To state the requirements in a way that allows tracing from the original (ambiguous, unrefined) form to the analyzed and refined form.

1.2 Intended Audience and Reading Suggestions

This document is not only for developers that need to create and innovate the product.

1.3 Product Scope

The product is a software application which is accessed from a web browser and used by people or companies to sell or buy houses or apartments.

1.4 References

- IEEE 830-1998
- Git repository : <https://github.com/bulbulica/wad>

2. Overall Description

2.1 Product Perspective

The “Online shop – real estate” is a virtually self-contained online shop; however, it will require users to have access to a web browser on their workstation computer. This means that the users of the system do not need to invest in any other software to get the most out of the software system as any Windows based PC comes installed with a web browser, and any non-Windows machine can use Firefox or other freeware browsers.

2.2 Product Functions

2.2.1 Reading offers

Without login every guest can view the offers with all information about a residence.

2.2.2 Login panel

Every guest can login with an existing trial or full account with a username and a password.

2.2.3 Register panel

It can be created a trial account using a username that is not already in the database and a password.

2.2.4 Add/Modify/Remove offer

While logged in you can add, modify or remove an offer record on the existing account.

2.2.5 Trial user limitations

A trial user has a limitation of a house/apartment sell.

2.2.6 Packages

Trial user can buy packages of 5\$, 25\$ or 40\$ and it will automatically become a full and get full account access.

2.2.7 Administrator rights

The administrator can add, close, modify or delete an offer record.
He is also responsible of the payment – accept or reject a payment from a user.

2.2.8 Send messages

All logged users can send messages to each other.

2.3 User Classes and Characteristics

Guests – He can only view the offers

Trial user – Add and manage only one offer and communicate with other accounts by sending messages.

Full user – Add and manage multiple offers, communicate with other accounts.

Administrator – All the rights, he can close, modify or delete an offer and he can accept or reject a payment.

2.4 Operating Environment

The product will work on every machine like: Windows, Linux or OS X that will have installed XAMPP or other similar product installed on it that will allow to access the HTTP and the MySQL panel. Therefore, no need of expensive hardware.

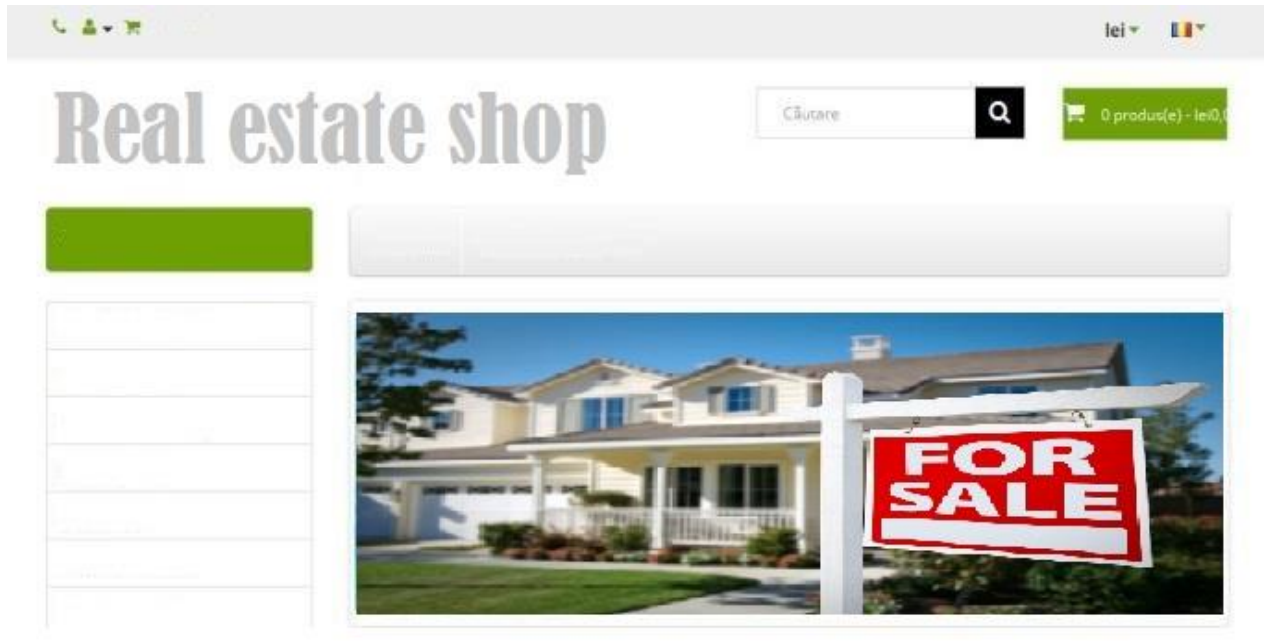
2.5 Design and Implementation Constraints

There are a number of constraints which the system must abide by during development. The system must be developed within their bounds. These constraints dictate a number of the functional and nonfunctional requirements specified by this document. All are important to be aware of during the implementation of the software system.

- System is to be developed for distributed use as a web application. This will limit the ability for real time updates to the system.
- System is to be developed in PHP.
- Data must be stored in a relational database for quick queries and storage.
- Passwords must be sent and stored in encrypted form.
- Some users are authorized users while some are non-authorized users.
- Packages of 5\$ can have only 5 offers, whereas the 25\$ can add a maximum of 20 offers and the last package of 40\$ can add a total of 50 offers
- Keep user overhead to an absolute minimum. Anywhere the system can handle a decision itself, it must do so.

2.6 Assumptions and Dependencies

- System will be added on a machine running Windows, Linux or OS X operating system, Apache and MySQL 5.0 or newer.
- An offer can be modified a maximum of 5 times in a day, if more, it will be considered spam and the offer will be deleted.
- Administrator will control the payment made by the users in case of scam.



- Trial accounts will stay only 3 months from the moments when they are created and they will be automatically deleted.
- Users will use browsers like Firefox, Chrome, IE, etc. without need of other external tools.

3. External Interface Requirements

3.1 User Interfaces

The design of the application will be close to the presented one and so the user will have the cart in right, the categories of houses or apartments in the left, some icons where he will get to his account and a bar where to search for a product.

3.2 Hardware Interfaces

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

3.3 Software Interfaces

It is preferred that the XAMPP to be under version 7.0, like 5.6.30 actual version on the apache website. If not, on Windows is accepted also Bitnami, for Linux is LAMP and for OS X is MAMP all of them under version 7.0 because we do not want further errors in our application. For the development of the application will be used PHPStorm IDE created by JetBrains PhpStorm because is a product that works perfectly with Symfony, Drupal, WordPress, Zend Framework, Magento and other frameworks. The writing standard will be told by the project manager.

3.4 Communications Interfaces

The application will communicate from port 80 (HTTP) or even on port 443 (SSL) and will need also the port 3306 (MySQL) for communication with the database.

4. System Features

4.1 Minimum front-end development

4.1.1 Description and Priority

A minimum part of front-end with HTML and CSS that will be the start of the application from where the back-end developers can create the real mechanism. Priority set as 4.

4.1.2 Stimulus/Response Sequences

Some pages, 5-10 HTML pages with content inside where will be some forms where the programmers can continue with the server part.

4.1.3 Functional Requirements

Index, contact, products, register, login, user panel, admin panel. Those are the important pages that have at least a minimum of HTML.

4.2 Minimum back-end development

4.2.1 Description and Priority

So the application has a front-end, but the important part is the back-end, the part where you can add/modify/delete offers and add/modify/delete products from the cart. Priority set as 7.

4.2.2 Stimulus/Response Sequences

From the existing HTML pages, the programmers need to create the PHP code that will execute and redirect you to whatever you desire: account, register, offers or cart.

4.2.3 Functional Requirements

- If the person logs in, the cookies is set for him, if he leaves and then he comes back he will have the same products in the cart.
- The register will check if a username is already in the database, if not, it will add a new record.
- In the login panel it will check if it the administrator, the paid user or even the trial user and it will redirect and login to the account where he entered the username and the password correctly.
- Add to cart, remove or modify the number of products that you want to buy will also be created in this back-end development session.

4.3 Improvements for the front-end interfaces

4.3.1 Description and Priority

The application is ugly and it needs a new and big improvement of the visual interface. JavaScript will be added here and add to card dynamically without refresh of the page, more CSS will be added to the style sheet and a better organization in the page of the existing HTML. Priority set as 5.

4.3.2 Stimulus/Response Sequences

Will be added JS in the offers panel, when a person adds a product in the cart, the page will not refresh.

4.3.3 Functional Requirements

- No page refresh when product added to cart.
- Better organization of the existing HTML.
- Bootstrap CSS will be added to the page.

4.4 Late back-end development

4.4.1 Description and Priority

Contact form will be added that contacts the administrator. Users can send each other private messages with a minimum of logged account even that is trial or full. If a person buys products from the cart, it will be able to add his credit card account and pay for the products. The administrator will get a notification that a payment is waiting to be accepted or rejected. Priority set as 5.

4.4.2 Stimulus/Response Sequences

A person can contact the administrator even that is logged in or not using the contact page.

4.4.3 Functional Requirements

- The contact form will work only for logged in users that want to contact the administrator.
- The private messages can be sent between every user even it is paid or not.
- Buy sequence where the user adds his account information to pay the cart.
- The administrator will be notified and he needs to accept or reject the pay.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The search query will be modified as they will limit only 10 results per page, this will make the response from the server faster.

5.2 Safety Requirements

In order to prevent scam or spamming the administrator will check every day the application and modify whatever it takes to prevent hacking or possible money loss from eligible people.

5.3 Security Requirements

In other to prevent SHELL upload, SQLI, CSRF or stored/reflected XSS, the upload of photos will be modified and checked if there is PHP, ASP or other programming codes inside, all the queries will be prepared, tokens will be used for every account and in the end every message will be sanitized.

6. Other Requirements

In the database, the user table and the offer table will be split in 2 or more, depending on how many people and offers will be on the application. It is very important that the response to the user to be delivered in a short amount of time.