

---

# **Software Requirements Specification**

**for**

## **‘Rent-A-Car’ Web Application**

**by Cristescu Ovidiu**

## Table of Contents

|   |   |
|---|---|
| 1. Introduction .....                               | 3 |
| 1.1 Purpose.....                                    | 3 |
| 1.2 Product Scope.....                              | 3 |
| 1.2 Intended Audience and Reading Suggestions ..... | 3 |
| 1.4 Definitions, Acronyms, and Abbreviations .....  | 3 |
| 1.5 References .....                                | 3 |
| 2. Overall Description .....                        | 4 |
| 2.1 Product Perspective .....                       | 4 |
| 2.2 Product Functions .....                         | 5 |
| 2.3 User Classes and Characteristics.....           | 5 |
| 2.4 Operating Environment .....                     | 5 |
| 2.5 Design and Implementation Constraints .....     | 5 |
| 2.6 User Documentation .....                        | 5 |
| 2.7 Assumptions and Dependencies .....              | 5 |
| 3. External Interface Requirements.....             | 6 |
| 3.1 User Interfaces.....                            | 6 |
| 3.2 Hardware Interfaces.....                        | 6 |
| 3.3 Software Interfaces .....                       | 6 |
| 3.4 Communications Interfaces.....                  | 6 |
| 4. System Features .....                            | 7 |
| 5. Other Nonfunctional Requirements.....            | 8 |
| 5.1 Performance Requirements .....                  | 8 |
| 5.2 Safety Requirements.....                        | 8 |
| 5.3 Security Requirements.....                      | 8 |
| 5.4 Software Quality Attributes.....                | 8 |
| 6. Other Requirements .....                         | 9 |
| Appendix A: Glossary .....                          | 9 |
| Appendix B: Analysis Models .....                   | 9 |
| Appendix C: To Be Determined List .....             | 9 |

## Revision History

| Name             | Date       | Reason For Changes | Version |
|------------------|------------|--------------------|---------|
| Cristescu Ovidiu | 24.03.2017 | First Revision     | 1.0     |

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to present a description of an online 'Rent-A-Car' business. It will explain the features of the system, the interface of the system and what the system will do.

## **1.2 Product Scope**

This application will be an online renting system. The system will be based on the client-server architecture.

## **1.3 Intended Audience and Reading Suggestions**

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

## **1.4 Definitions, Acronyms, and Abbreviations**

Not Applicable

## **1.5 References**

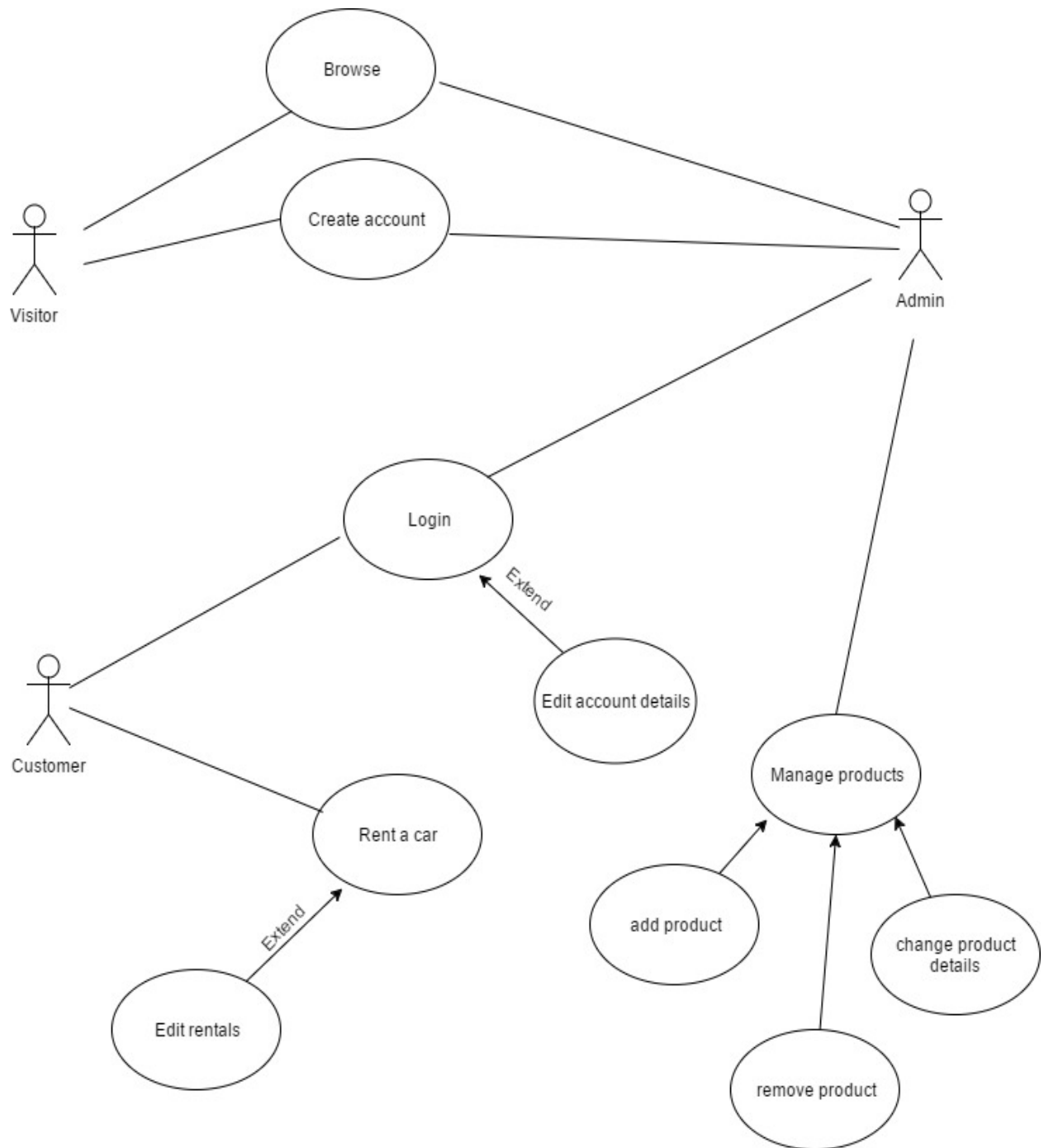
Git repository: <https://github.com/cristee95/>

## 2. Overall Description

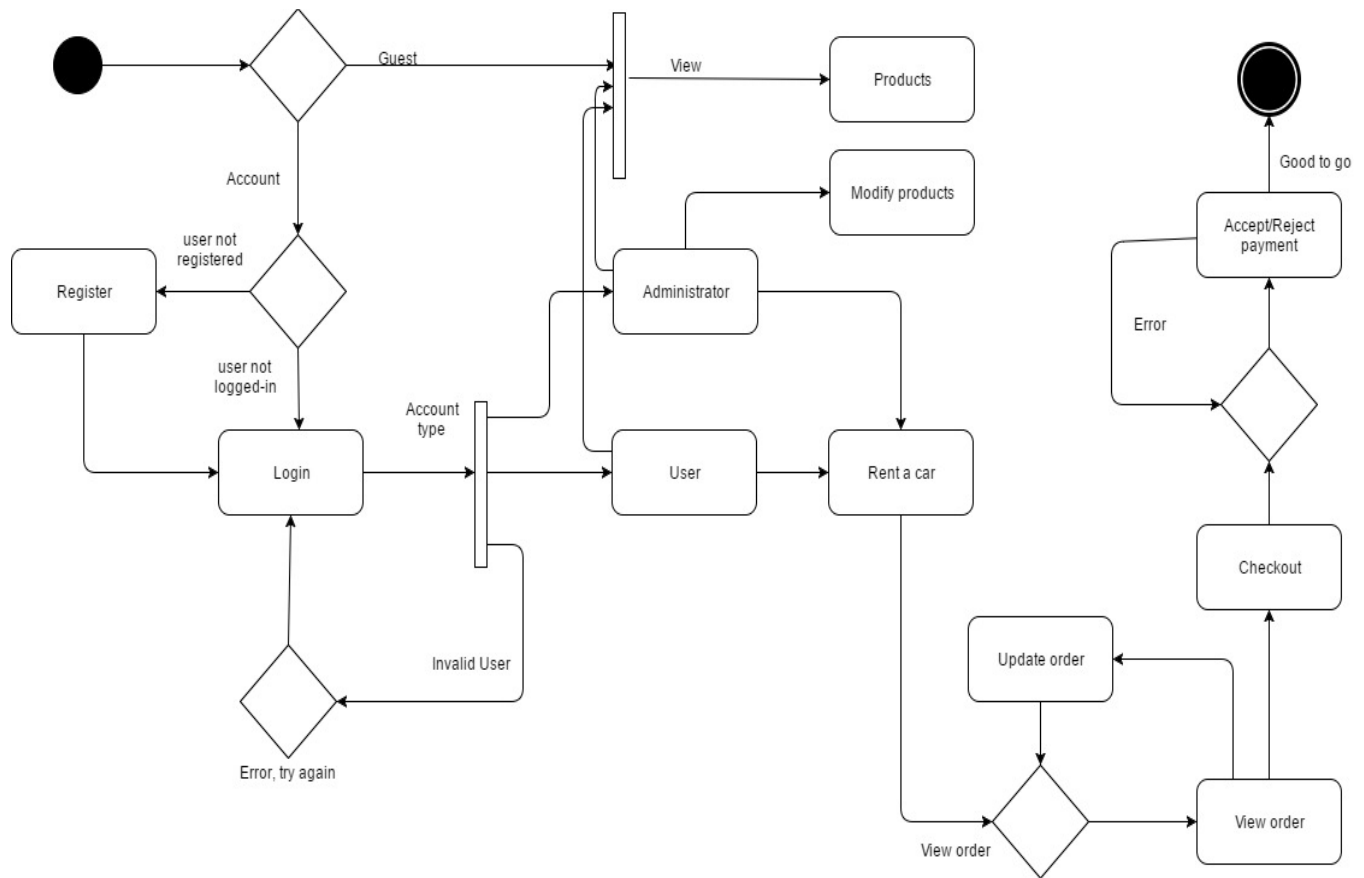
### 2.1 Product Perspective

This application is self-contained product which will allow administrators to manage the database. Regular user permissions will include browsing products and placing orders.

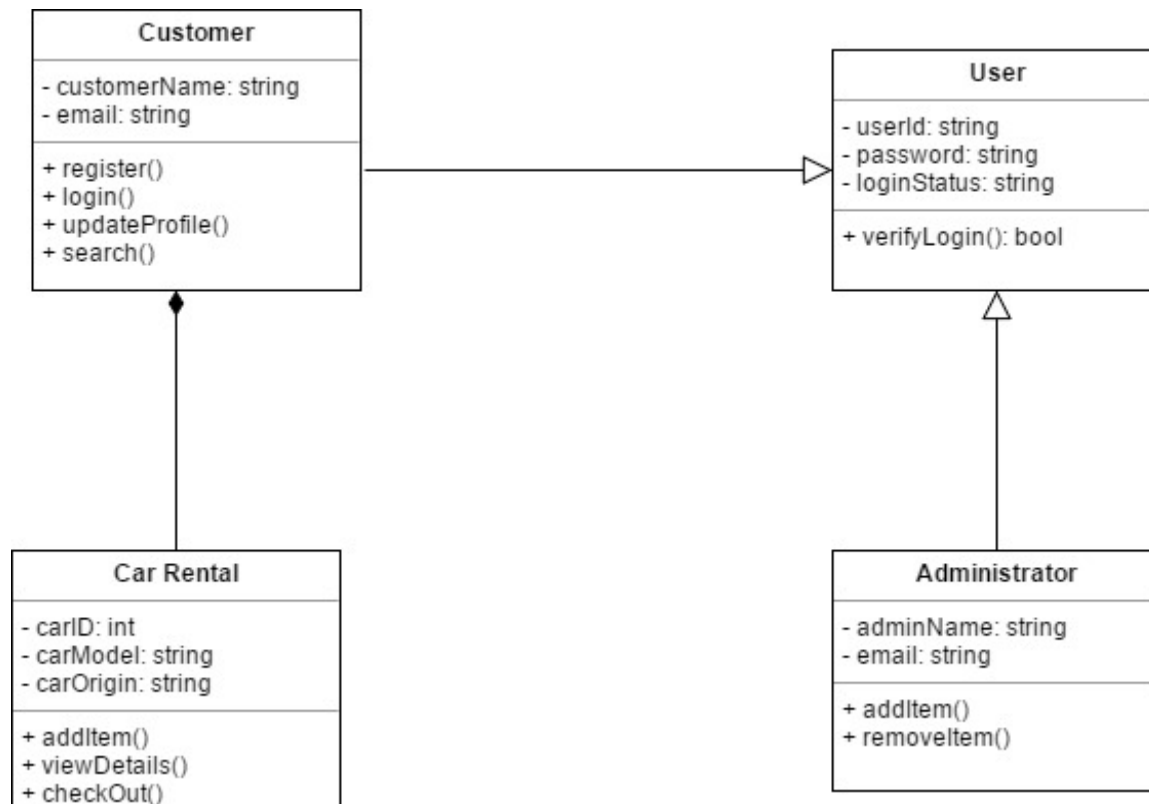
#### Case diagram



## Activity diagram



## Class diagram



## **2.2 Product Functions**

The main functionalities of the application will be:

- Authentication
- View product list
- Managing products (administrators only)
- Managing account details

## **2.3 User Classes and Characteristics**

Users of the application will be people who have an interest in renting a car.

## **2.4 Operating Environment**

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

## **2.5 Design and Implementation Constraints**

The application will be displayed only in English.

Frontend: HTML, CSS, JavaScript

Technology: PHP

## **2.6 User Documentation**

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

## **2.7 Assumptions and Dependencies**

The main prerequisites include: the user having an active Internet connection and accessing the application only from compatible browsers.

When filling in a form, the users must respect the formatting options of each field, otherwise error/warning messages will be displayed and the action won't be completed.

## 3. External Interface Requirements

### 3.1 User Interfaces

The application general layout will contain:

- The header which displays:
  - the logo
  - a search bar
- The body, which displays content
- The footer with:
  - Contact
  - Delivery methods
  - Copyright

The user interface will contain:

- A menu bar
- A list with cars (with prices and pictures)

The administrator interface will contain:

- A list with products and options for each of them
- Forms for changing product details or adding new items to the database

### 3.2 Hardware Interfaces

Not applicable

### 3.3 Software Interfaces

The application will be connected to a database which stores details about cars, as well as user accounts.

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

Transfer protocol used: HTTP.

## 4. System Features

- 4.1.1 Description and Priority
- 4.1.2 Stimulus/Response Sequences
- 4.1.3 Functional Requirements

### Functional requirements:

- **Authentication**
  - *Log in* using user id (email) and password
  - *Forgot password* option for resetting password, in case the user does not remember his password
  - *Create new account* for people who want to purchase products
- **View product list**
  - *Search for specific item*
  - *Search by name and brand*
  - *Order products by name or price*
  - *See details.*
- **Managing products (administrators only)**
  - *Add items* to inventory
  - *Delete items* from inventory
  - *Edit product details (such as picture, price or description)*
- **Managing account details**

Both regular users and administrators will be allowed to change their personal account details (name, address, telephone number).



## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

### 5.2 Safety Requirements

Not applicable

### 5.3 Security Requirements

Personal data is stored in the database (and encrypted).

### 5.4 Software Quality Attributes

**1. Reliability:** personal information (such as address) will be kept safe. The transactions, which will have delayed results (shipped days or weeks later), will be safe procedures.

**2. Usability:** The application will have a broad customer base, therefore using it shall be as simple as shopping at a store. It will meet the customers' need to be able to use the web site without training. The software will flow according to the users' expectations, offer only needed information (when needed).

**3. Security:** Customer data and other electronic information will be handled as securely as possible.

**4. Availability:** The application will have "24/7" availability, and it will be operational every day of the year. Availability means that the website will be and running 24/7/365 and it will be available when accessed by diverse compatible browsers.

### 5.5 Business Rules

**Disclaimer** Sellers should make sure that their items are authentic before listing them on the website. Sellers may not disclaim knowledge of, or responsibility for, the authenticity or legality of the items they offer in their listings. If a seller cannot verify the authenticity of an item, it shouldn't be listed on the website.

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.

## **6. Other Requirements**

### **Appendix A: Glossary**

### **Appendix B: Analysis Models**

The strategy adopted for this project is the Agile approach

### **Appendix C**