Rent a car application

Use-Case Model

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <02April/2019> | <1.0> | First version of the document | Olar Paul Dan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-Cases Identification 4

2. UML Use-Case Diagrams 5

Use-Case Model

# Use-Cases Identification

We can identify three actors, each having different privileges: the common customer, the car renting company’s employee and the administrator.

The user (the car renting company’s customer): (requires a User Name and a Password) in order to have access to the car database, accessing the car’s information and renting a car. Once a new account is created, the new user has access to all of this features, he will be able to search and view the available cars (detailed car information and pictures) for the selected date interval and from the specified source city. To rent a car, the customer must fill its personal data. After the customer has rented a car, an email is sent to him with the renting details.

The car renting’s company employee: (requires a username and password) in order to access his/her full set of privileges. When a customer that has rented a car comes to the renting company to pick up the car, an employee creates a contract which is then printed so that the client can sign it. When a car is brought back by a customer, the employee marks the car as returned.

The admin: has the highest set of privileges when using this application. The admin also requires a password and a correct user name (unique) to have access to the information about the cars and users. He/she is able to add access to a user, delete an account, delete a user from the database, check the current accounts and modify the interest for a selected account. In other words what he/she is also able to do is CRUD operations over cars, customers and employees.

**Use case: logging in**

**Level: employee-goal level**

**Primary actor: Employee, Customer, Administrator**

**Main success scenario: - enter username and password**

**- pressing the login button**

**Extensions: login incorrect, message prompted**

**Use case: check available cars**

**Level: user-goal level**

**Primary actor: Customer**

**Main success scenario: -after a successful login, customer accesses a frame containing a list/table with all the available cars.**

**-user inputs the date interval and the source city**

**-the system filters down the cars and displays only the ones that correspond to the inserted query**

**Extensions: the selected date interval or city doesn’t have any car to display, message prompted**

**Use case: Renting a car**

**Level: user-goal level**

**Primary actor: Customer**

**Main success scenario: -the user selects the desired car.**

**-system prompts for the desired date interval for renting and the city from which the customer will come to pick up the car.**

**-customer fills in his/her personal data.**

**-system sends an email to the customer containing the renting detailed information.**

**Extensions: the email inserted by the customer is not valid, the use case fails since the renting information cannot be sent.**

**Use case: CRUD operations on customers**

**Level: admin-goal level**

**Primary actor: Administrator**

**Main success scenario: -the administrator logins to the system using his/her unique credentials**

**-the administrator performs the CRUD operations through the interface that is provided to him by the system**

**Extensions: None**

# UML Use-Case Diagrams

