Software Requirements Specification

for

Booking Car Management System

Version 1.0 approved

Prepared by 0367604729

09/03/2021

Table of Contents

[1. Introduction 1](#_Toc66112579)

[1.1 Purpose 1](#_Toc66112580)

[1.2 Document Conventions 1](#_Toc66112581)

[1.3 Project Vision and Product Scope 1](#_Toc66112582)

[1.3.1 Vision Statement 1](#_Toc66112583)

[1.3.2 Product Scope 1](#_Toc66112584)

[1.4 References 1](#_Toc66112585)

[2. Overall Description 1](#_Toc66112586)

[2.1 User Classes 1](#_Toc66112587)

[2.1.1 Guest 1](#_Toc66112588)

[2.1.2 Authenticated User 1](#_Toc66112589)

[2.1.3 Customer 2](#_Toc66112589)

[2.1.4 Partner 2](#_Toc66112589)

[2.1.5 System Handler 2](#_Toc66112589)

[2.1.6 Administrator 2](#_Toc66112589)

[2.2 Usecase diagram 3](#_Toc66112590)

[2.3 Operating Environment 4](#_Toc66112591)

[2.4 Design and Implementation Constraints 4](#_Toc66112592)

[2.5 User Documentation 4](#_Toc66112593)

[2.6 Assumptions and Dependencies 5](#_Toc66112594)

[2.7 ER Diargam 5](#_Toc66112594)

[3. Functional Requirements / Use case details 6](#_Toc66112595)

[3.1 Use Cases for Guest 6](#_Toc66112596)

[UC-G-1: Login to system 6](#_Toc66112597)

[UC-G-2: Register 6](#_Toc66112597)

[UC-G-3: View list cars 7](#_Toc66112597)

[UC-G-4: View information details of a car 7](#_Toc66112597)

[UC-G-5: Search cars 7](#_Toc66112597)

[3.2 Use Cases for Authenticated User 8](#_Toc66112602)

[UC-AU-1: Logout 8](#_Toc66112603)

[UC-AU-2: Change password 8](#_Toc66112603)

[UC-AU-3: Manage profile 8](#_Toc66112603)

[3.4 Use Cases for Customer 8](#_Toc66112609)

[UC-C-1: Add car to cart 8](#_Toc66112610)

[UC-C-2: View cart 8](#_Toc66112610)

[UC-C-3: Manage cart 8](#_Toc66112610)

[UC-C-4: Add discount to order 9](#_Toc66112610)

[UC-C-5: Choose payment 9](#_Toc66112610)

[UC-C-6: Checkout 9](#_Toc66112610)

[UC-C-7: Track booking 9](#_Toc66112610)

[UC-C-8: View history 10](#_Toc66112610)

[UC-C-9: Search history 10](#_Toc66112610)

[3.4 Use Cases for Associated Company 10](#_Toc66112609)

[UC-AC-1: Manage car 10](#_Toc66112610)

[UC-AC-2: Manage discount 11](#_Toc66112610)

[UC-AC-3: Get the booking from the customer 11](#_Toc66112610)

[UC-AC-4: Manage customer 12](#_Toc66112610)

[UC-AC-5: Manage order 12](#_Toc66112610)

[3.4 Use Cases for System Notification](#_Toc66112609) 13

[UC-S-1: Notify customers about available discounts 13](#_Toc66112610)

[UC-S-2: Remind booking date and time to customers 13](#_Toc66112610)

[UC-S-3: Remind customer’s booking to associated company 13](#_Toc66112610)

[UC-S-4: Notify feedback to asscoiated company 13](#_Toc66112610)

[3.3 Use Cases for Administrator 13](#_Toc66112606)

[UC-A-1: Manage account 13](#_Toc66112607)

[UC-A-2: Manage feedback 14](#_Toc66112607)

[4. External Interface Requirements 14](#_Toc66112616)

[4.1 User Interfaces 14](#_Toc66112617)

[4.1.1 Standard website 15](#_Toc66112619)

[4.1.2 Standard phone application 15](#_Toc66112619)

[4.1.3 Login interface 15](#_Toc66112619)

[4.1.4 Register interface 15](#_Toc66112619)

[4.1.5 Search 15](#_Toc66112619)

[4.1.6 Cart view 15](#_Toc66112619)

[4.1.7 Admin’s control panel 15](#_Toc66112619)

[4.1.8 Associated company’s control panel 15](#_Toc66112619)

[4.2 Software Interfaces 15](#_Toc66112618)

[4.3 Communications Interfaces 15](#_Toc66112619)

[5. Other Nonfunctional Requirements 15](#_Toc66112620)

[5.1 Performance Requirements 15](#_Toc66112621)

[5.2 Security 15](#_Toc66112622)

[5.3 Localization 15](#_Toc66112623)

[5.4 Maintainability 15](#_Toc66112623)

[5.5 Legal and Copyright 15](#_Toc66112623)

[6. Business Rules 15](#_Toc66112624)

# Introduction

## Purpose

This document describes the requirements for Booking Car Management System, which supports 3rd company upload car information publicly. In addition to, helping customer book cars easily and convenient.

## Document Conventions

The functional requirements for this project are organized by use case within use case with user class.

## Project Vision and Product Scope

### Vision Statement

For people who are interested in traveling, Booking Car Management System is an application that will provide information about the cars which were provided by 3rd company, allow customers to book a car in various types, maybe in the future, we will expand our list of partners, customer segments and business areas.

### Product Scope

Booking Car Management System is an application that will provide information about the cars which were provided by 3rd companies. It will provide fulfillment of car orders placed on application and track the orders.

# Overall Description

## User Classes

### Guest

Guest is someone who has not yet logged into the system. If guests are in the phone application, they can see the list of cars as well as see the details of each type of car and search for different types of cars. To be able to place an order, they need to login if they already have an account or register if they have not. If guests are in the website, they have to login first to do the actions.

*UC-G-1: Login to system*

*UC-G-2: Register*

*UC-G-3: View list cars*

*UC-G-4: View information details of a car*

*UC-G-5: Search cars*

### Authenticated User

Authenticated User is a user who logged into system and had every functions that a guest can do. The difference is that they have logged into the system. At this moment, they can log out if they want, change password and manage their profile like avatar or edit their information only for associated company and customers.

*UC-AU-1: Logout*

*UC-AU-2: Change password*

*UC-AU-3: Manage profile*

### Customer

Customer is a authenticated user using phone application or website, they can search cars, add more car into cart, view cart, manage cart, apply discount, choose the way to pay or checkout cart to order. After order, they can track their booking and view their history orders.

*UC-C-1: Add car to cart*

*UC-C-2: View cart*

*UC-C-3: Manage cart*

*UC-C-4: Apply discount to order*

*UC-C-5: Choose payments*

*UC-C-6: Checkout*

*UC-C-7: Track booking*

*UC-C-8: View history*

*UC-C-9: Search history*

### Associated Company

Associated Company is an authenticated user using website, they shall manage cars and manage discounts such as add, edit and remove those items. After that, they shall provide them to admin. In additions, they shall be able to get the booking from the customers to approve or deny and manage their customer also the customer’s orders.

*UC-AC-1: Manage car*

*UC-AC-2: Manage discount*

*UC-AC-3: Get the booking from the customer*

*UC-AC-4: Manage customer*

*UC-AC-5: Manage order*

### 2.1.5 System Notification

System Notification is responsible for suggesting available discounts to customers and informing customers when their rental date is about to take place. For associated company, inform them of order information as well as feedback coming from customer who rent their car, reminding them to deliver the car to the customer.

*UC-S-1: Notify customers about available discounts*

*UC-S-2: Remind booking date and time to customers*

*UC-S-3: Remind customer’s booking to associated company*

*UC-S-4: Notify feedback to associated company*

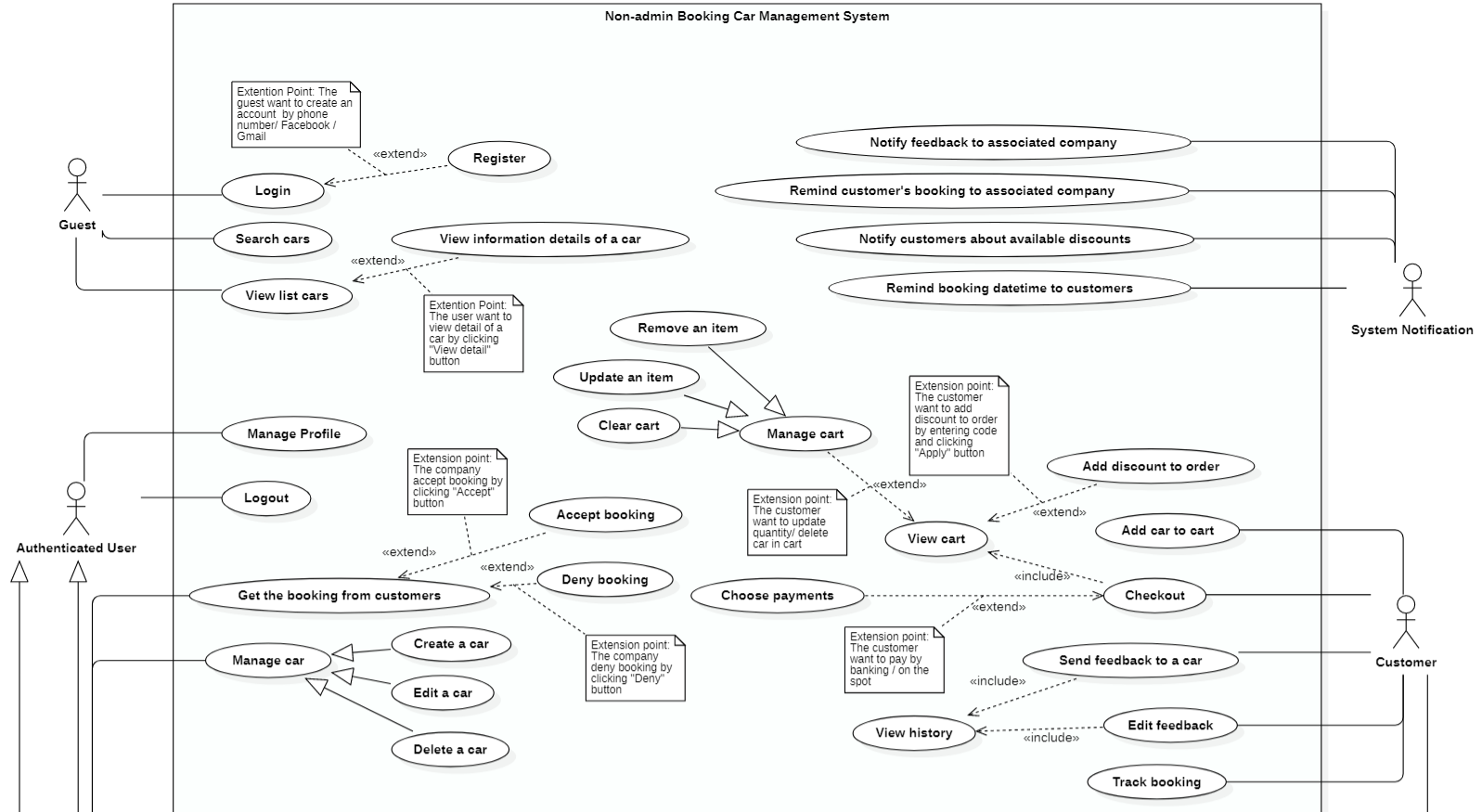
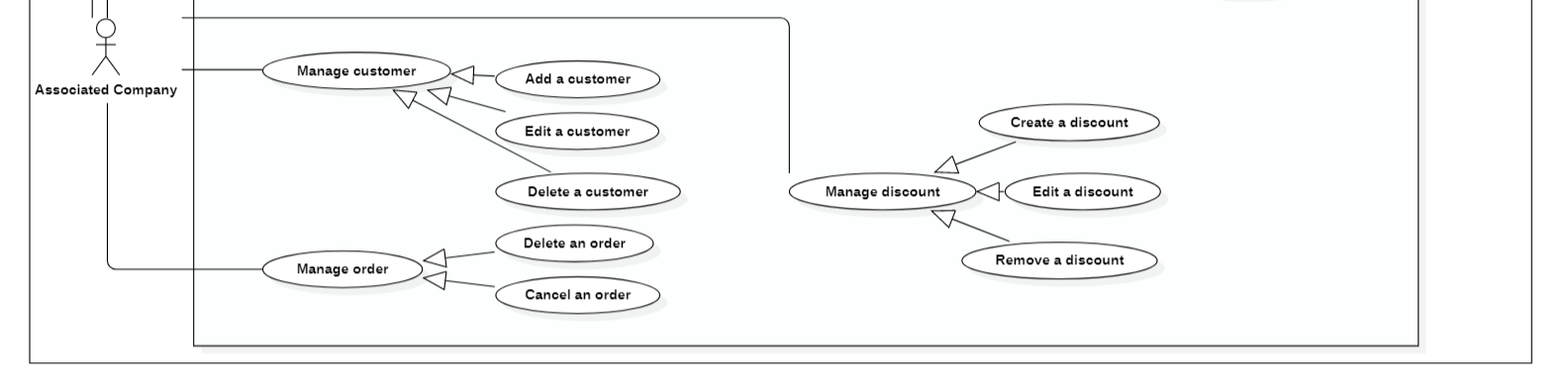
### 2.1.6 Administrator

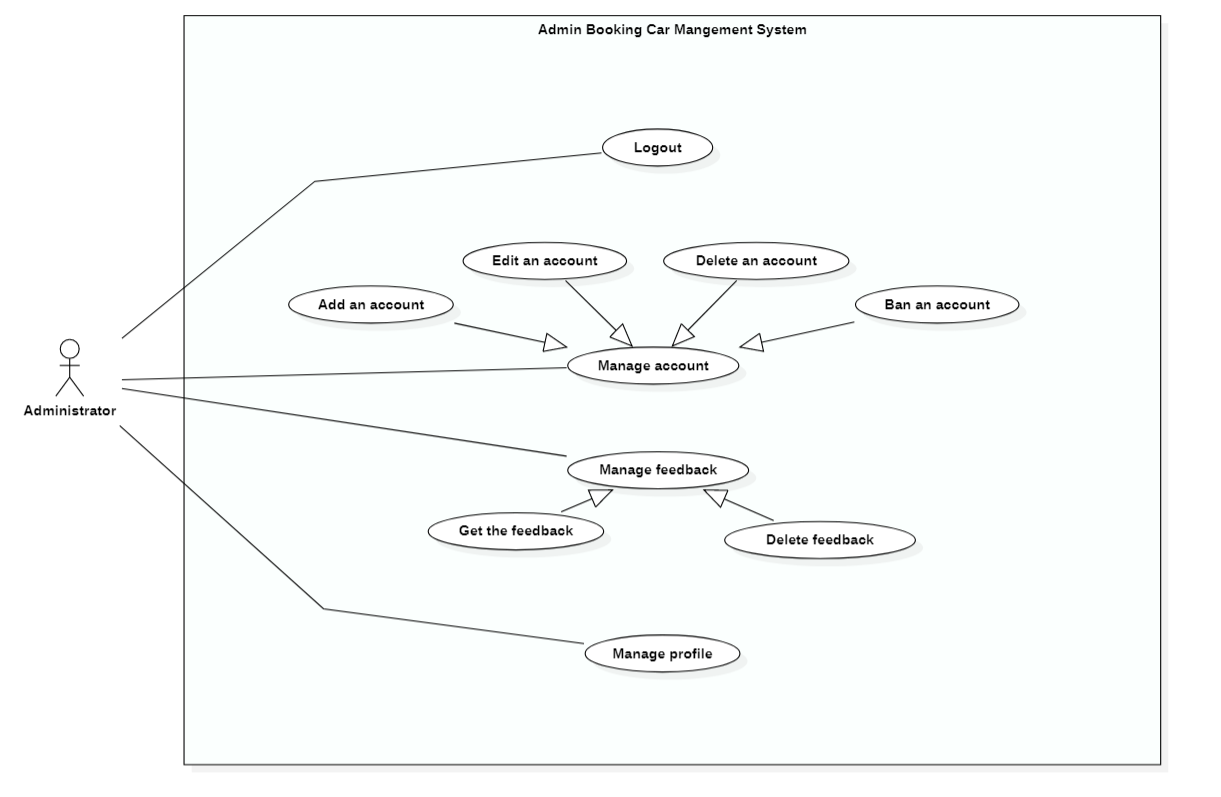
Administrator is an authenticated user, they can manage accounts and feedbacks from the customer.

*UC-A-1: Manage account*

*UC-A-2: Manage feedback*

## Use case diagram

2.2.1 Non-admin Booking Car Management System

2.2.2 Admin Booking Car Management System

## Operating Environment

Booking Car Management System for Website shall be compatible with standard web browsers on Windows, Mac, and Web-enabled smart phone platforms.

(Supported browsers are Internet Explorer 5, 6, 7, and 8; Firefox 2 and 3; Opera 10 and 11; Google Chrome 10 and later.)

Booking Car Management System for Phone Application shall be compatible with Android and IOS platforms.

(Supported Android 5.0 and later; IOS 8.0 and later.)

## Design and Implementation Constraints

CO-1: The Administrator shall be able to maintain and modify the site without requiring any specialized technical knowledge beyond basic HTML or the use of site editing tools provided by a commercial web site builder, such as WordPress.

CO-2: The website shall be developed using the Java platform and design using HTML, CSS, and JS.

CO-3: The phone application shall be developed using Android Studio with Java Language.

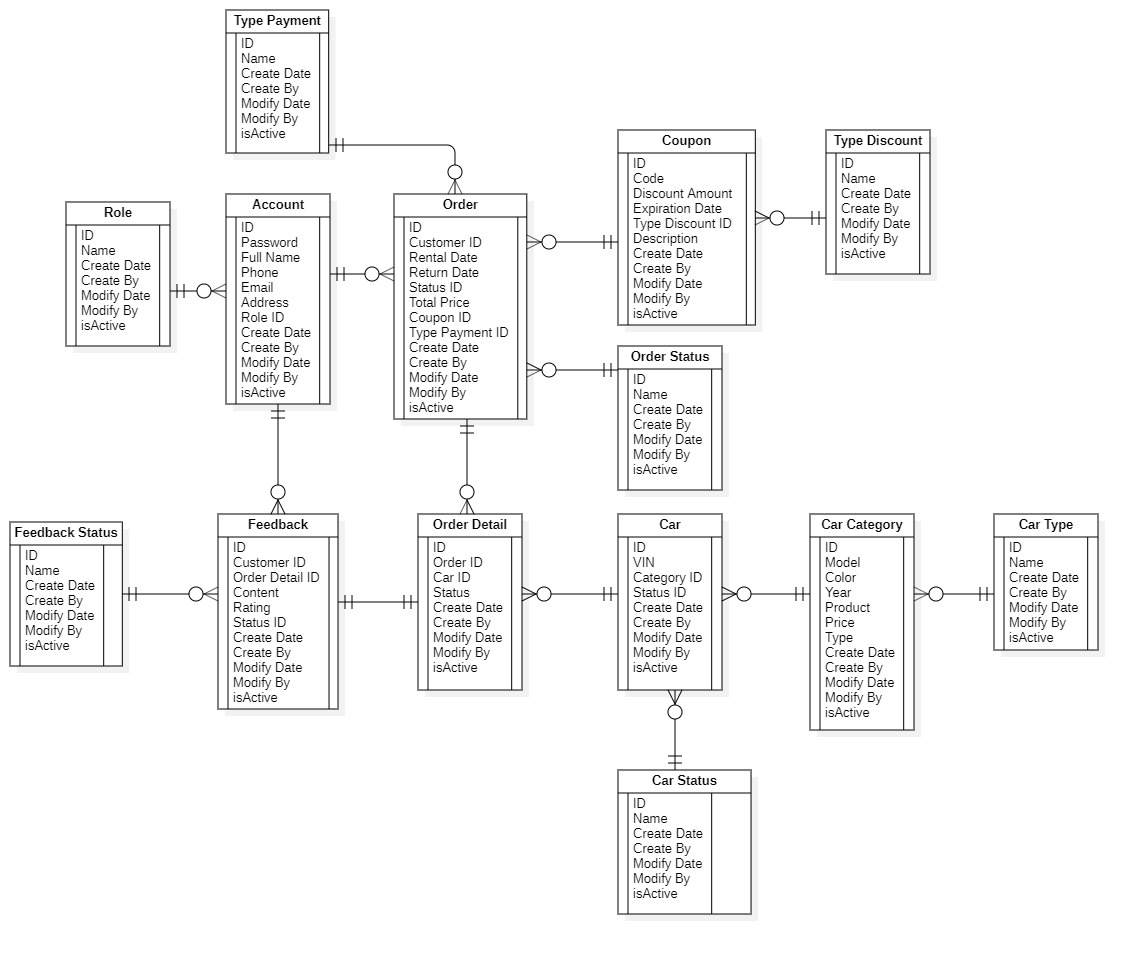
## User Documentation

The developer shall provide necessary documentation to enable an Administrator to perform the use cases listed in section 2.1.4 Administrator.

## Assumptions and Dependencies

There are no known assumptions or dependencies for this website and mobile app.

## ER Diagram



# 

# Functional Requirements / Use case details

*Notes: - H: High*

*- M: Medium*

*- L: Low*

## Use Cases for Guest

UC-G-1: Login to system

Description: A guest shall be able to login to the system to do more actions.

|  |  |  |
| --- | --- | --- |
| login.form: | A Guest shall be able to login to the system by entering their email address and password that available in system accounts list to the form. | (Priority=H) |
| login.method: | A Guest shall be alle to login by connecting with a facebook account or a gmail account. | (Priority=H) |
| login.recaptcha: | A Guest shall check re-CAPTCHA to confirm that is human being. | (Priority=H) |
| login.recaptcha.error: | If a Guest don’t check re-CAPTCHA, the website shall display an error message. | (Priority=H) |
| login.valid.account: | The application shall display an error message if the Guest enters invalid email address or password that is not available in system account list. | (Priority=H) |
| login.redirect.verify: | If a Guest enters an account that is inactive, the application shall redirect the guest to a verify page. | (Priority=H) |
| login.verify: | The Guest shall entering a code that system has sent to their email. | (Priority=H) |
| login.verify.error: | If a Guest enters an invalid verified code, the application shall display an error message. A code may be invalid because the code doesn’t match the code in a guest’s email. | (Proprity=H) |
| login.redirect.home: | The application shall redirect a user to home if a user entering valid account that has been verified and already checked re-CAPTCHA. After login, the user may be a customer or an admin depend on their role and their using enviroment. | (Priority=H) |

UC-G-2: Register

Description: A guest shall be able to create a new account in the system.

|  |  |  |
| --- | --- | --- |
| register.form: | A Guest shall be able to create new account by entering email address, password and confirm password and other information such as firstname, lastname, address, phone. | (Priority=H) |
| register.valid.email: | The application shall display an error message if the e-mail address that the Guest enters is missing or not properly formatted. A valid e-mail address contains only letters and numbers, must contain exactly one “@” following one or more characters, must contain exactly one period following the “@” with any number of intervening characters, and must have at least one character following the period. | (Priority=H) |
| register.valid.password | The application shall display an error message if the password is not matched with the proposed format of the application, that would be more than 6 characters and included: word, number and special characters (like @, #, $...) | (Priority=H) |
| register.valid.confirm: | The website shall display an error message if a guest doesn’t entering the confirm password that doesn’t match the password. | (Priority=H) |
| register.valid.field: | The website shall display error messages if any text fields are empty. | (Priority=H) |
|  | The application shall display an error message if the guest enters invalid email address or password that is not available in system account list. | (Priority=H) |
| register.method: | A Guest shall be alle to sign up by connecting with a facebook account or a gmail account. | (Priority=H) |
| register.send: | The application shall send the verified code via email to a guest’s email. | (Priority=H) |
| register.redirect.login | The application shall redirect a guest to login form if a guest creates new account successfully. | (Priority=H) |

UC-G-3: View list cars

|  |  |  |
| --- | --- | --- |
| list\_cars.view: | A Guest that on customer site shall be able to view list available twenty cars order by created date at home page at the first time access the application. | (Priority=M) |

UC-G-4: View information details of a car

|  |  |  |
| --- | --- | --- |
| car.detail.view: | A Guest that on customer site shall be able to view detail of a car such as car name, seats, feedbacks,… | (Priority=H) |

UC-G-5: Search cars

|  |  |  |
| --- | --- | --- |
| car.search: | A Guest that on customer site shall be able to search available cars by entering car name or choosing car type, car quantity, rental date and return date. | (Priority=H) |

## Use Cases for Authenticated User

UC-AU-1: Logout

Description: An Authenticated User shall be able to logout application.

UC-AU-2: Change password

Description: An Authenticated User shall be able change his password. The application shall display an error message if he entered the wrong previous password or invalid new password field.

*Constraint: The valid password must be more than 6 characters and contains at least one word, one number and one special character (like @, #, $ …)*

UC-AU-3: Manage profile

Description: An Authenticated User shall be able to manage his profile.

|  |  |  |
| --- | --- | --- |
| profile.avatar.add: | An Authenticated Usershall be able to add an avatar to profile. | (Priority=L) |
| profile.add.error: | If an Authenticated User add an image that bigger than 900 x 900, the application shall display an error message. | (Priority=L) |
| profile.information.modify: | An Authenticated User shall be able to modify profile information such as name, address, phone. | (Priority=M) |
| profile.modify.error: | The application shall display error messages if user enters empty text fields. | (Priority=M) |

## Use Cases for Customer

UC-C-1: Add car to cart

Description: A Customer shall be able to add more than one available car to cart.

UC-C-2: View cart

Description: A Customer shall be able to view cart.

UC-C-3: Manage cart

Description: The application shall use a shopping cart-based on-line store mechanism that permits a Customer to remove cars from the cart, clear the cart, update quantity of a car, recalculate each subtotal and the total price of the cars in the cart, else he/she can continue booking.

|  |  |  |
| --- | --- | --- |
| cart.car.remove: | A Customer shall be able to remove cars from the cart | (Priority=H) |
| cart.clear: | A Customer shall be able to clear all the cars from the cart | (Priority=H) |
| cart.car.update: | A Customer shall be able to increase or decrease the quantity of a car in the cart. | (Priority=H) |
| cart.price.recalculate: | If a Customer update quantity of a car, the calculating mechanism shall re-calculate each subtotal of that car or if a Customer remove cars or update quantity of cars in the cart then calculating mechanism shall re-calculate again total price of the cars. | (Priority=H) |
| cart.continue: | A Customer shall be able to redirect to home page to book more cars. | (Priority=L) |

UC-C-4: Add discount to order

Description: A Customer shall be able to add available discount to reduce the total price.

|  |  |  |
| --- | --- | --- |
| discount.add: | The shopping cart mechanism shall provide a discount code field that provides a specific discount percentage amount. Allowable discount percentages shall range from 10% to 100%. | (Priority=H) |
| discount.error: | If a Customer enters an invalid discount code, the application shall display an error message. A code may be invalid because the code is not known by the application, it has expired, or it doesn’t apply to any of the cars currently in the cart. | (Priority=H) |

UC-C-5: Choose payments

Description: A Customer shall be able to choose available payments to order.

UC-C-6: Checkout

Description: A Customer shall be able to checkout cart to completely order.

|  |  |  |
| --- | --- | --- |
| checkout.confirm: | The application shall display an confirmation box with the order informations such as customer’s name, address, phone, the order detail list and total price. | (Priority=H) |
| checkout.error: | If any car in order detail aren’t avalable or not enough car quantity at the time customer checkout then the application shall display an error message. | (Priority=H) |
| checkout.redirect: | If the order is being set succesffully then redirect Customers to thank you page with their order information. | (Priority=H) |
|  |  |  |

UC-C-7: Track booking

Description: A Customer shall be able to track the booking before it reaches him/her.

|  |  |  |
| --- | --- | --- |
| booking.phases: | A Customer shall see the phases from booking to delivery included: Waiting for confirmation, Confirmed, Preparation, Delivery | (Priority=H) |
| booking.details: | A Customer shall view details of the booking. | (Priority=H) |
| booking.profile: | A Customer shall be able to see the profile of associated company. | (Priority=M) |
| booking.cancel: | A Customer shall be able to cancel the order before the delivery phase | (Priority=H) |

.

UC-C-8: View history

Description: A Customer shall be able to view history of previous orders.

|  |  |  |
| --- | --- | --- |
| history.orders: | A Customer shall view the list previous orders which are completed. | (Priority=H) |
| history.detail: | A Customer shall view detail of a order. | (Priority=H) |
| history.booking\_again: | A Customer shall be able to book again a car in the order. The car shall be added to customer’s cart. | (Priority=M) |
| history.booking\_again.error: | The application shall display an error message if the booked car is out of stock or not available in the application. | (Priority=M) |
| history.feedback: | A Customer shall be able to send the feedback of each car by entering the content and rating from 1 to 10. | (Priority=M) |
| history.feedback.edit: | A Customer shall be able to edit the feedback of a car in each order detail by modifying the content or rating from 1 to 10. | (Priority=M) |
| history.feedback.error: | The application shall display an error message if the content field is empty and rating stars are not checked. | (Priority=M) |
| history.profile: | A Customer shall be able to see the profile of asscoiated company. |  |
| history.profile.report: | A Customer shall be able to report an associated company by entering the reason content or choose many from each available reasons in the list. | (Priority=M) |
| history.profile.report.error: | The application shall display an error message if the reason content field is empty or available reason aren’t checked. | (Priority=M) |

UC-C-9: Search history

|  |  |  |
| --- | --- | --- |
| history.search: | A Customer shall be able to search available orders by entering car name or order date. | (Priority=H) |

## Use Cases for Associated Company

UC-AC-1: Manage car

Description: An Associated Company shall be able to create a new car for the on-line store, edit and delete the existing car.

|  |  |  |
| --- | --- | --- |
| car.create: | An Associated Company shall be able to create a new product for the on-line store. Attributes of a car are a car name, feature, type, color, seats, description, image, price and status. | (Priority=H) |
| car.edit: | An Associated Company shall be able to edit any of the attributes of an existing car in the on-line store. If the status’s checkbox is checked, the car’s status is changed to “Activate”; | (Priority=H) |
| car.valid | The application shall display an error message if any field is empty when creating or modifying. |  |
| car.delete: | An Associated Company shall delete the existing car in the list. A confirmation box will appear to get the confirm of him before executing the delete action. Delete a car is just about change the status of that car to “Deactivate”. | (Priority=H) |
| car.feedback: | An Associated Company shall be able to view the feedback list of each car, it includes customer name, customer’s image, feedback content and stars. | (Priority=L) |

UC-AC-2: Manage discount

Description: An Associated Company shall be able to create, edit and remove the discount code that provides the specific percentage for specified cars.

|  |  |  |
| --- | --- | --- |
| discount.create: | An Associated Company shall be able to create a discount for the specific cars in the list. Attributes of a discount are discount name, percent, start date and expiry date, the status of discount and what cars it applied for. | (Priority=H) |
| discount.edit: | An Associated Company shall be able to edit any of the attributes of an existing discount. If the status’s checkbox is checked, the discount’s status is changed to “Activate”; | (Priority=H) |
| discount.valid: | The application shall display an error message when an Associated Company entered the invalid values, that not matched with those constraints:  - Discount Name must be more than 2 characters and contains only word and number.  - Percent must be an integer and from 10 to 100.  - Start date must be before Expiry date and after the current date. |  |
| discount.remove: | An Associated Company shall remove the existing discount in the list. A confirmation box will appear to get the confirm of him before executing the delete action. Delete a discount is just about change the status of that discount to “Deactivate”. | (Priority=H) |

UC-AC-3: Get the booking from customer

Description: An Associated Company shall be able to get the booking from customer to accept or deny.

|  |  |  |
| --- | --- | --- |
| booking.detail: | An Assoicated Company shall view the detail of the booking. | (Priority=H) |
| booking.customer\_profile: | An Associated Company shall see the profile of a customer by a link the application provided. | (Priority=M) |
| booking.confirm: | An Associated Company shall make the confirmation to accept or deny the booking from customer. | (Priority=H) |

**UC-AC-5: Manage customer**

Description: An Associated Company shall be able to manage customer such as add, edit and delete action. If they do not have own the customer management system, the website shall provided them a URL to manage their own booking customers. However, if they have their own customer management system, they can connect to their system through API integration (There is the API that synchronize data between the customer’s system and our system). In additions, they can see the profile of each customer and report that customer if they want.

|  |  |  |
| --- | --- | --- |
| customer.add: | An Associated Company shall be able to add a customer to their own list if a customer executes booking cars on their car list. Attributes of a customer are customer’s username, fullname, phone number and address. | (Priority=H) |
| customer.edit: | An Associated Company shall be able to edit the information of a customer. | (Priority=H) |
| customer.delete: | An Associated shall be able to delete an existing customer. The confirmation box shall appear to get the confirm before deleting. | (Priority=H) |
| customer.valid: | The application shall display an error message when an Associated Company entered the invalid values, that not matched with those constraints:  - Customer Name must not be empty.  - Phone number must be valid Vietnam phone.  - Address must not be empty. | (Priority=H) |
| customer.profile: | An Assoicated Company shall be able to view the profile of each customer. | (Priority=H) |
| customer.report: | An Associated Company shall be able to report an customer by entering the reason content or choose mmany from each available reasons in the list. | (Priority=M) |
| custome.report.error: | The application shall display an error message if the reason content field is empty or available reason aren’t checked. | (Priority=M) |

**UC-AC-6: Manage order**

Description: An Associated Company shall be able to manage the order such as view order list, delete the order and archive a set of orders by customers. The order will automatic add into the list by the system, any editing is not allow.

|  |  |  |
| --- | --- | --- |
| order.delete: | An Associated Company shall delete an existing order in the list. The confirmation box shall appear to get the confirm before deleting. |  |
| order.detail: | An Associated Company shall view the detail of orders in the list by a link the application provided. | (Priority=M) |
| order.status: | The application shall show the status of order that it is one of the several states: Not Yet, Renting, Returned. | (Priority=M) |
| order.cancel: | An Associated Company shall be able to cancel the order. The confirmation box shall appear to get the confirm before cancel the order | (Priority=M) |

## Use Cases for System Notification

UC-S-1: Notify customers about available discounts

Description: An System Handler shall notify Customers when they access the application about many available discounts.

UC-S-2: Remind booking date and time to customers

Description: An System Handler shall be able to remind Customers when they access the application about their booking date and time until the current date is the booking date if Customers have made the orders.

UC-S-3: Remind customer’s booking to associated company

Description: An System Handler shall be able to remind Associated Company when they access the application about customer’s booking date and time until the current date is the booking date if customers have made the orders.

UC-S-4: Notify feedback to associated company

Description: An System Handler shall notify Associated Company when they access the application about many feedbacks from customers.

## Use Cases for Administrator

UC-A-1: Manage account

Description: An Administrator shall be able to view the list of accounts to see account’s information. Besides, thay shall add, edit, delete or just ban the account from the list.

|  |  |  |
| --- | --- | --- |
| account.add: | An Administrator shall be able to add an account to the system. | (Priority=H) |
| account.edit: | An Administrator shall be able to edit an existing account. | (Priority=H) |
| Account.delete: | An Administrator shall be able to delete an existing account | (Priority=H) |
| account.type: | The website shall divide accounts according to each type of user. | (Priority=H) |
| account.profile: | An Administrator shall be able to view the profile of each accounts. | (Priority=H) |
| account.ban | An Administrator shall be able to ban an account if an account has not been used for more than one year or an account violated something in terms of use. | (Priority=M) |

UC-A-2: Manage feedback

Description: An Administrator shall be able to get and delete the feedback from customer.

# External Interface Requirements

## User Interfaces

**4.1.1 Standard website**

In case the users are using application on the website, includes: Customer, Associated Company and Admin. These are the following interface similarities:

Must have a header with navigation bar witth View Cart, Login and Register link items.

Must have a footer with show the informations such as: Address, Road map, Website copyright information, Link to social networking sites and menu.

Header and footer must have the same color and should not be too dark to see.

Content of the website must be centered of the screen.

**4.1.2 Standard phone application**

Must have a bottom bar incluse the items such as Home, Search, Cart and User, when selecting one of the items, that item must be focused by bigger font size, bold color,…

There will be a pop-up box to show the detail of car when clicking to the image or the name of a car or the profile of user when clicking to the image or the name of a user and the background behind may be blurred.

When displaying a list of cars, only the photos and names should be displayed to not take up too much space, as well as the image is small but enough to show the user what to see.

**4.1.3 Login interface**

When asks the user to type his username and password, if the user entered either his username and password incorrectly then an error message occurs.

Must have the link to register page.

**4.1.4 Register interface**

When the user type information details, if the user entered the input values have not validate, then error message shall be shown below the fields.

Must have the link to login page.

**4.1.5 Search**

The customer or associated company can enter the book name he is looking for or select the type of car to search. Besides, application must have a calendar to select the rental date and the return date to search cars available between that dates.

Car Type must have the icon to classify the type of cars.

**4.1.6 Cart view**

Cart view shows the booking car with ability to Customer to remove, update quantity and apply discount to the booking car.

- Must have an area to change the order date.

**4.1.7 Admin’s control panel:**

This control panel will allow admin to add, remove or ban users; to get and delete feedback from the customer.

**4.1.8 Associated Company’s control panel:**

This control panel will allow company to add, edit and remove car; add, edit and remove discounts; And manage the orders from customers.

## Software Interfaces

Login with Facebook or Google.

External payment system shall be Visa, Master Card, Napas.

## Communications Interfaces

There are no specific communications interfaces besides URL links.

# Other Nonfunctional Requirements

## Performance Requirements

The web pages shall fully paint in an average response time of 5 seconds or less over a broadband (DSL or cable) Internet connection.

**5.2 Security**

The website shall use standard Web security protocols (HTTPS/SSL 3.0) when transferring any private information regarding a Guest, Customer, or Associated Company and order information of customer.

**5.3 Localization**

The application shall support and money conversion which show on the screen.

**5.4 Maintainability**

The code must be clear and match the correct syntax and rule in Java language.

**5.5 Legal and Copyright**

Not allow to use image or information of associated company for external distribution.

# Business Rules

BR-1: When a Customer adds a car to cart, he has to login to the system first.

BR-2: A Customer can only cancel the order when the rental date of the order greater than the current date.

BR-3: A Customer can feedback to a car in the order when and only when the order is complete.

BR-4: When a customer choose the rental date and the return date of a order, he must ensure that the return date must be after the rental date and the rental date must be after the current date.

BR-5: A Customer has to return all the car in the order on time when it is the drop off date.

BR-6: Coupon must be valid and available when entered.