Software Requirements Specification

for

<Easy Car Project>

EASYCAR

Version 1.0 approved

Prepared by

<Eyad Khaled Qbori>
<Aseel Sami Ahmad>
<Abdelkarim Al-Zahrani>
<Mohammed Al-Basri>

<Um Al-Qura university>

<4/29/2020>

Table of Contents

Revision History

Name	Date	Reason for Changes	Version

1. Introduction

1.1 Purpose

It's a broker system between customers and cars owners, so that car owners can add their own cars in the system and customers can make a rent request on the cars in nearest place and in any time.

1.2 Product Scope

This software will help customer to rent a car easily from the system, also the car owners can accept or reject the rent request coming from the customers, This software applies an automated system to take the rent from the customer directly to the car owners, so that provide a new way of car renting.

The product scope of this project is:

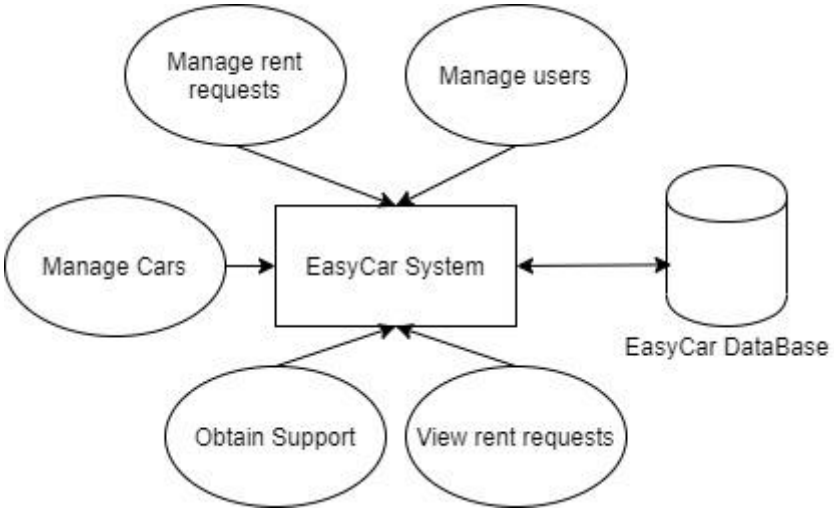
1- Facilitate renting cars for customers
2- Make additional income for car owners
3- Facilitate the mobility idea
4- Make the best prices for the customers
5- view rent requests easily
6- Schedule rent request in any time
7- Cancel any rent request in a less than 24 hours
8- Easy payment
9- Obtain customers support

1.3 Intended Audience and Reading Suggestions

The document is intending for all stakeholders and the developers (Customers, Car owners and the admins), The shall have a basic knowledge in SRS and other rental car system,
Knowledge and understanding of use case diagrams also required.

2. Overall Description

2.1 Product Perspective



Easy Car is a public system for all people, the main aim of the system is to facilitate renting cars by best prices for customers in an innovative and modern way, so that you can rent cars from anywhere and anytime from the system. Also, customers can view their rent requests easily, and car owners can view rent requests on their cars.

2.2 Product Functions

1- Login and logout from the system
2- Schedule / cancel a rent request for customers
3- View rent requests to customers
4- Make a payment for rent requests
5- Display an invoice for the rent request
6- Add / Remove / Change cars statues easily
7- Display the rent request that's coming from customers
8- Obtain customers support by receiving problems and solve it
9- The admin can mange customers, car owners and support employees

2.3 User Classes and Characteristics

We have three types of users in this system can access:

Admin: This user has an ability to Block user that has too much reports, also he can obtain support to other users by receiving problems and trying to solve it.

Customers: This user has an ability to Rent request to car shown in the system, or cancel the request easily, also this user can view the rent requests.

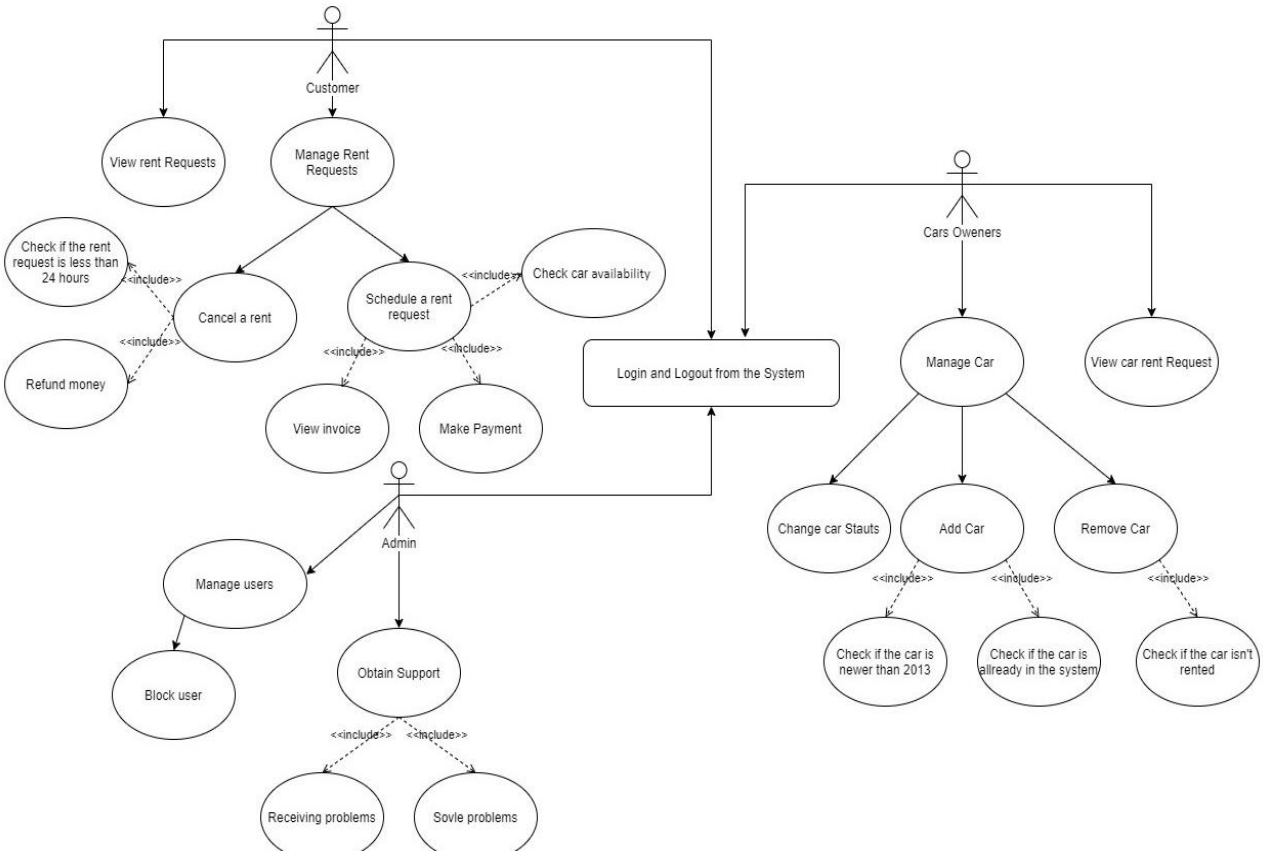
Car owners: This user has an ability to add or remove car to the system and he can change the car statues easily, also he can view the rent requests that coming from customers.

2.4 Operating Environment

This system will be work on any browser that can have an internet connection
Operating system: Windows 10, Mac OS X, Linux, android, iOS
Database: MySQL
Platform: HTML 5, CSS, JavaScript

3. Use case

3.1 Use case diagram



3.2 Use case description

3.2.1.1 Schedule rent request

Use case:	Schedule rent request
Summary:	The user enters the start date, end date, pick up location and the system show up a list of cars available then the user will choose a car from the list then he will make a payment to confirm his request.
Basic course of events:	1- The system will ask the user to enter the start date and the end date for the rent request 2- The user should enter the start and end date for the rent request 3- The system will show up a map and ask the user to choose a pickup location 4- The user will choose the pickup location from the map 5- The system will check the cars available for rent 6- The system will show up a list of cars available for renting 7- The system will check if the user has a valid driver's license 8- The user will choose from the available cars in the list 9- The system will ask the user to choose a time for pick up 10- The user enters the time he desired 11- The system will ask the user to make a payment for his rent request 12- The user will make the payment for the rent request 13- The system will ask the user to confirm the rent request 14- The user should confirm the rent request 15- The system will show up the rent invoice after payment
Alternative paths:	In step 1,3,5,9 and 13 the user can return to the homepage and exit the system
Exception points:	In step 3, if the user chooses a location that out of pick up range, the system will show up a message that says, "This location is out of service range", and go back to step 2. In step 11, if the user doesn't payment enough money the system automatically will decline the rent request and go back to step 9.
Trigger:	The user chooses schedule rent request from the menu.
Assumptions:	The user shall be more than 18 years old and valid driver's license
Precondition:	None
Post condition:	The confirm message will show up to the user for his rent request

3.2.1.2 Cancel a rent request

Use case:	Cancel a rent request
Summary:	The user shall choose a rent request from a list within 24 hours then the system will cancel it and refund money to the user.
Basic course of events:	1- The system views a list of rent requests 2- The user will choose a rent request from the list 3- The system check if the rent request within 24 hours 4- The system asks the user to confirm his cancelation request 5- The user will confirm his cancelation request 6- The system will refund the money to user
Alternative paths:	In step 2,5 the user can return to the homepage and exit the system
Exception points:	In step 3, if the user chooses a request after 24 hours the system will decline his cancelation request
Trigger:	The user chooses cancel rent request from the menu.
Assumptions:	none
Precondition:	none
Post condition:	The confirm message will show up to the user for his cancelation request

3.2.1.3 View rent requests

Use case:	View rent requests
Summary:	The user can choose a rent request from the menu.
Basic course of events:	1- The system displays a menu 2- The user will choose his request from the menu 3- The user shall wait to accept or reject his rent request 4- The system will show his rent request details
Alternative paths:	In step 2 the user can return to the homepage or exit the system
Exception points:	In step 3, if the car owner rejects the request and go back to step 2.
Trigger:	The user chooses cancel rent request from the menu.
Assumptions:	none
Precondition:	none
Post condition:	None

3.2.2.1 Obtain support

Use case:	Obtain support
Summary:	The user can receive any problems from other users and solve it.
Basic course of events:	1- The system shows the support menu 2- The user will choose the problem from the list 3- The system will show the problem details 4- The user will communicate with the problem sender and solve.
Alternative paths:	In step 2 and 4 the user can return to the homepage or exit the system
Exception points:	None
Trigger:	The user chooses obtain support from the menu.
Assumptions:	None
Precondition:	None
Post condition:	None

3.2.2.2 Block user

Use case:	Block user
Summary:	The system will show a list then the user chooses from the list
Basic course of events:	1- The system shows a list of users 2- The user will choose the Clint “Car owner or Customer” from the list 3- The system will show the Clint details 4- The user will block the Clint of using the system 5- The system updates the clients list
Alternative paths:	In step 2 and 4 the user can return to the homepage or exit the system
Exception points:	None
Trigger:	The user chooses block user from the menu.

Assumptions:	None
Precondition:	None
Post condition:	None

3.2.3.1 Add car

Use case:	Add car
Summary:	The user can add his car to the system easily by enter his car information: “Company, name, size, color, plate number, model, price for one day and an image “, Then the system adds the car.
Basic course of events:	1- The system asks the user to enter car information 2- The user enters car information into the system 3- The system check if the car is newer than 2013 4- The system check if the car is already registered 5 - The system asks the user to confirm his add request 6- The user confirms his add request
Alternative paths:	In step 2 and 6 the user can return to the homepage or exit the system
Exception points:	In step 2, if the user leaves an empty filed, the system will decline his add request and back to step 1. In step 3, if the car is older than 2013 the system will decline his add request and go back to step 1. In step 4, if the car is already registered, the system will decline his add request and go back to step 1.
Trigger:	The user chooses to add a car from the menu.
Assumptions:	The car is newer than 2013 The car isn’t having any damage
Precondition:	none
Post condition:	The confirm message will show up to the user for his add request

3.2.3.2 Remove car

Use case:	Remove car
Summary:	The user chooses his car from a list to remove it then the system removes the car.
Basic course of events:	1- The system shows a list of cars and asks the user to choose 2- The user will choose a car from the list 4- The system will check if the car is already rented 5 - The system asks the user to confirm his remove request 6- The user confirms his remove request 7-The system update his car list
Alternative paths:	In step 2 and 6, The user can cancel his remove request and exit the system.
Exception points:	In step 4, if the car is already rented the system automatically decline his remove request and shows message says, “The is already in use” and go back to step 2.
Trigger:	The user chooses a remove request from the menu.
Assumptions:	The car isn’t rented to complete the task
Precondition:	None
Post condition:	Conformation message shows to the user

3.2.3.3 Change car statues

Use case:	Change car statues
Summary:	The user chooses his car from a list to change its statues then the system updates the car the list.
Basic course of events:	1- The system shows a list of cars and asks the user to choose 2- The user will choose a car from the list 3 - The system asks the user to change the car statues 4- The user chooses the statues his desired to “Available” or “Under service” 5- The system will check if the car is already rented or not 6-The system asks the user to confirm the change request 7-The user confirm his change request 8-The system update his car list
Alternative paths:	In step 2, 4 and 7 The user can cancel his change request and exit the system.

Exception points:	In step 5, if the car is already rented the system automatically decline his remove request and shows message says, “The is already in use” and go back to step 2.
Trigger:	The user chooses a change statues from the menu.
Assumptions:	The car isn’t rented to complete the task
Precondition:	None
Post condition:	Conformation message shows to the user

3.2.3.4 view rent requests

Use case:	view rent requests
Summary:	The system displays a menu and the user shall choose to “accept” or “reject” the rent request from the customers.
Basic course of events:	1- The system displays a list 2- The user will choose the rent request from the menu 3- The user shall choose to “accept” or “reject” the rent request from customers 4- The system asks the user to confirm the rent request 5- The user confirms the rent request 6- The system will update the list
Alternative paths:	In step 2,4 and 5, The user can cancel and exit the menu.
Exception points:	None
Trigger:	The user chooses a view rent request from a menu.
Assumptions:	None
Precondition:	None
Post condition:	Conformation message shows to the user

3.2.4.1 Sign in

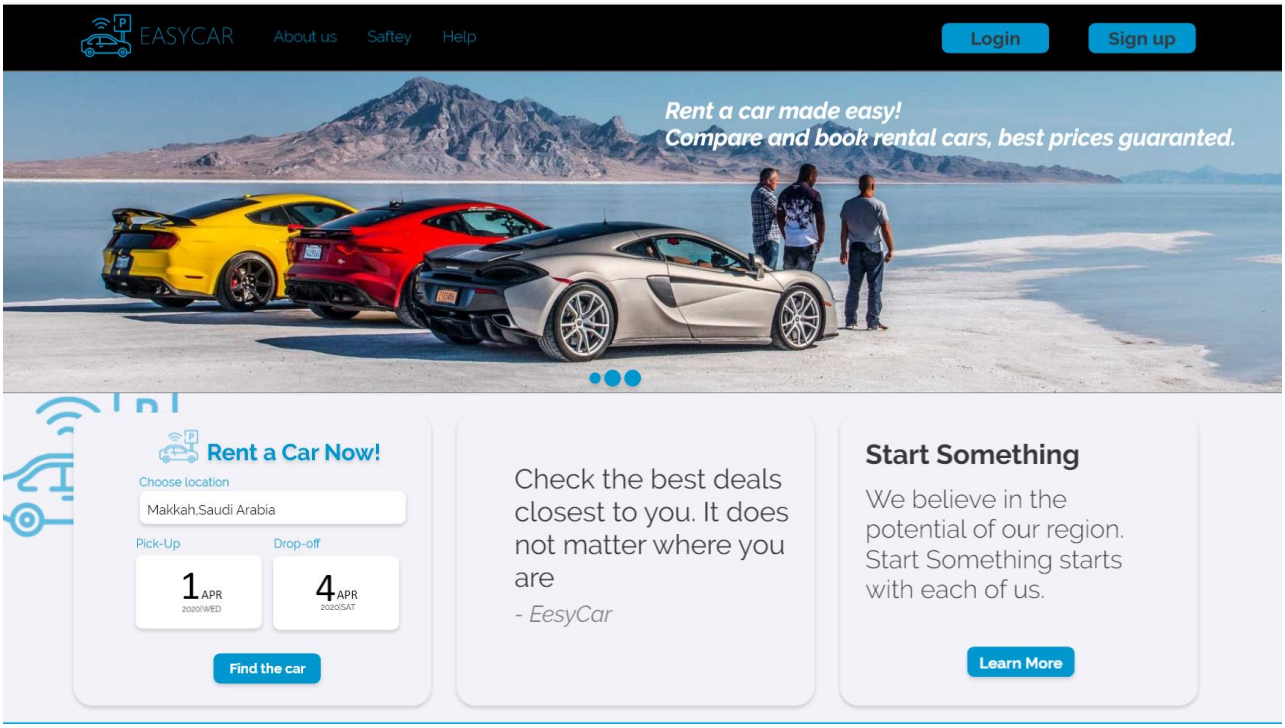
Use case:	Sign in
Summary:	The user chooses Sign in from the menu, the system will sign in the user to the system
Basic course of events:	1- The system asks the user to enter his information 2- The user enters his email and password 3- The system check if the user already registered 4- The user choose sign in button to login 5- The system shall sign in the user to his dashboard as "Car owner, customers or Admin"
Alternative paths:	In step 2,4 The user can cancel and exit the menu
Exception points:	None
Trigger:	The user chooses a view sign in from a menu.
Assumptions:	User already registered
Precondition:	None
Post condition:	None

4. External Interface Requirements

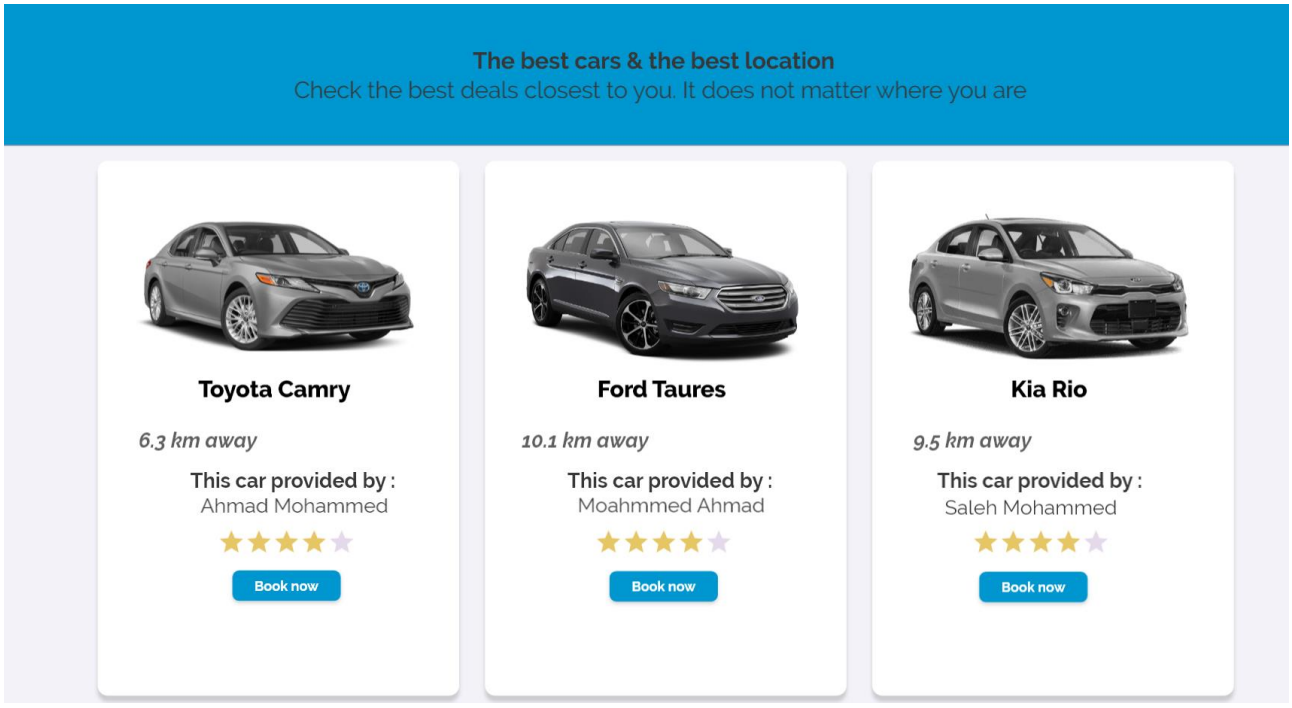
4.1 User Interfaces

EasyCar system uses English language as default be easy to use and understandable, our website uses a friendly interface.

Home Screen



Rent request screen



4.2 Hardware Interfaces

There are no required hardware interfaces needs for this system.

4.3 Software Interfaces

This system is required a database to read and write the data about customers, owners and cars.

4.4 Communications Interfaces

This system is required a "https" protocol and Wi-Fi connection, to communicate between devices.

4.5 Functional Requirements

4.5.1 Manage Rent Requests

4.5.1.1 Schedule rent request

Requirement ID:	R01.1	Use case:	Schedule rent request
Description:	The system shall allow the user to schedule a rent request, the rent request will be allowed for the costumers if the owner accept his request. -if the user leaves aa date empty the system will decline his rent request And show a decline message. -if the user enters an invalid date the system shall show an error message -if the user tries to make a rent request on same car, the system shall show an error message.		
Rationale:	This will provide an ability for the user to schedule the rent requests and enjoy the car in the time he entered.		
Inputs:	Start date, end date and pick up location		
Outputs;	A conformation message. An invoice.		
Persistent change:	The rent request will be scheduled in the costumer list.		
Related requirement:	None		

4.5.1.2 Check car availability

Requirement ID:	R01.2	Use case:	Check car availability
Description:	The system shall check if the availability of a car in the selected time The car is available if: 1-Has no rent requests in the selected time 2-Any change of car statues from car owners		
Rationale:	To avoid duplicate of rent on the same car available on the list, so the system shall check if it available or not.		
Inputs:	Start date, End date, pick up location		
Outputs;	Message says the is "available or not available "		
Persistent change:	None		
Related requirement:	R0.1.1		

4.5.1.3 Make Payment

Requirement ID:	R01.3	Use case:	Make Payment
Description:	The system shall check time the user entered to the system, if it available then it will take the user to payment screen to pay and continue the rent request. -if the user has an invalid card the system shall show a decline payment message. -if the user hasn't enough balance in his credit card the decline payment message will show on the screen. -if the user leaves an empty filed the system shall show a decline message shown on the screen		
Rationale:	Without payment method you can't continue the rent request, the money will transfer to the car owner automatically		
Inputs:	Start date, end date, the name on the card		
Outputs;	a conformation message view invoice.		
Persistent change:	The payment saved in the car owner wallet		
Related requirement:	R01.2		

4.5.1.4 Cancel rent request

Requirement ID:	R01.4	Use case:	Cancel rent request
Description:	The system shall allow user to cancel his rent request within 24 hours and refund the money his card. -if the user leaves an empty filed for his cancel request, the system shall decline his cancel request and show an error message. -if the user trying to cancel the rent request after 24 hours, the system will decline his cancel request automatically.		
Rationale:	Allow the user to cancel the rent request for any reason, ex: if the car has any damage he can cancels easily, or if the car owner hasn't disclosed about any problem, then he can refund the money from the car owner.		
Inputs:	Start date, end date, reason for cancel.		
Outputs;	Conformation message. Refund the money		
Persistent change:	The car reruns available in the list of rent requests		
Related requirement:	R01.6		

4.5.2 View rent Requests

Requirement ID:	R02.0	Use case:	View rent Requests
Description:	The system shall allow the user to view his rent requests for the costumer in the main menu. -if the user enters an invalid rent request, the system shall show an error message.		
Rationale:	Allow the users to track the rent request from the system easily in the list.		
Inputs:	Start date, end date,		
Outputs;	The rent request for the user		
Persistent change:	None		
Related requirement:	None		

4.5.3 Manage users

4.5.3.1 Block user

Requirement ID:	R03.1	Use case:	Block user
Description:	The system shall allow the user or car owner to send a report about any problem - If more than one report is sent to a user, the user must be block from using the program		
Rationale:	The user must submit any problem in order to deal with the problem correctly		
Inputs:	Reports		
Outputs;	Show message to the user blocked says “You’re no longer to use the system anymore”		
Persistent change:	The user can’t access the system		
Related requirement:			

4.5.4 Obtain Support

Requirement ID:	R04.0	Use case:	Obtain Support
Description:	In order to receive problems from the user and searches for a solution to those problems		
Rationale:	-It must be present in the systems that deal with users in order to receive problems from them and provide them with solutions		
Inputs:	Sending problems and complaints through the system		
Outputs;	Shows solutions and fixes through the system to the users		
Persistent change:	none		
Related requirement:			

4.5.5 Manage Car

4.5.5.1 Add car

Requirement ID:	R05.1	Use case:	Add car
Description:	The system shall allow the car owner to add his car for rent. -If the car older than 2013 he cannot add the car it will display error massage.		
Rationale:	Allow the car owner to add his car for hire and earn good money		
Inputs:	Plate Number, car color, car model, car Type.		
Outputs;	Allow the car owner to add his car.		
Persistent change:	The system will display a message indicating that the car has been added.		
Related requirement:			

4.5.5.2 Remove car

Requirement ID:	R05.2	Use case:	Remove car
Description:	The system shall allow the car owner to remove his car. -If the car is rented, he cannot remove the car it will display error massage.		
Rationale:	Allow the car owner to remove his car if something wrong		
Inputs:	Plate Number, car color, car model, car Type.		
Outputs;	Allow the car owner to remove his car.		
Persistent change:	The system will display a message indicating that the car has been remove.		
Related requirement:			

4.5.5.3 Change car statues

Requirement ID:	R05.3	Use case:	*Change car statues*
Description:	The system shall allow the user to change statues of the car changes from available to reserved or Conversely		
Rationale:	-In order to clarify for all users in the system from the knowledge of the available cars, which can be a rental option -Clarification that some cars are not ready for rental, meaning they are pre-booked or broken down, and so on		
Inputs:	Cars information		
Outputs;	The statues of car cases on the site as available or Out of service		
Persistent change:	-If the car is available, it will join the list of available cars -If it is out of service it will disappear from the system		
Related requirement:			

4.5.7 Sign in

Requirement ID:	R07.0	Use case:	Sign in
Description:	The system shall allow the user to sign in by checking his email and his password if he registered in the system or not -if the user leaves an empty filed for his cancel request, the system shall decline his cancel request and show an error message. -if the user doesn't register in the system an error message will show on the screen -if the email or password is wrong an error message will show on the screen.		
Rationale:	With out sign into the system you can't access the system and you can't use any Features.		
Inputs:	Email, password		
Outputs;	Display the wrong password message		
Persistent change:	None		
Related requirement:	None		

4.6 Performance Requirements

High data transfer rate.
Uninterrupted interrupted connections
It must be able to handle Advertisement shown on the website

4.7 Non-Functional Requirements:

1. Usability: Touch screens, keyboard and mouse.
2. Efficiency: Easy should function on the same system
3. Security: only the users registered have the authorization.
   The system shall automatically log out the user after a period of time.
4. Reliability: information about cars or rent requests should be secured and only accessed by attendees.
5. Accuracy: The system shall be accurate in terms of scheduling.
6. Scalability: The system shall be built in a way that new functionalities can be added easily.