Online Car Dealer System

Professor

Team Members

Table of Contents

[1. Introduction 4](#_Toc380084786)

[2. Current system 5](#_Toc380084787)

[3. Proposed system 6](#_Toc380084788)

[3.1 Overview 6](#_Toc380084789)

[3.2 Functional requirements 6](#_Toc380084790)

[3.3 Nonfunctional requirements 6](#_Toc380084791)

[3.4 Constraints (“Pseudo requirements”) 6](#_Toc380084792)

[3.5 System models 6](#_Toc380084793)

[3.5.1 Scenarios 6](#_Toc380084794)

[3.5.2 Use case model 6](#_Toc380084795)

[3.5.3 Object model 6](#_Toc380084796)

[3.5.3.1 Data dictionary 6](#_Toc380084797)

[3.5.3.2 Class diagrams 6](#_Toc380084798)

[3.5.4 Dynamic models 6](#_Toc380084799)

[3.5.5 User interface 6](#_Toc380084800)

[4. Glossary 7](#_Toc380084801)

# 1. Introduction

(Johann)

# 2. Current system

Ask Professor what should go here.(Johann Henao)

# 3. Proposed system

## 3.1 Overview

(Johann)

## 3.2 Functional requirements

1- The system shall have a function to search cars from the available car inventory. The search criteria will be through selectable values, no free format input from the user.

2 – The system shall have a function to route a user to register or login.

3 - The system shall display the search result (from requirement 1) with the basic car information.

4 - The system shall allow the user to request a more detailed car information per car within the search results from functional requirement 3, which will return a page with pictures of the car, basic car information and additional notes.

5 - The system shall have a function to purchase a car. To purchase a car, a user must be registered and logged in.

6 – The system shall allow a user to register and/or login.

7 – The system shall allow a user to reset the login password in case it is forgotten.

8 – The system shall display an order confirmation after the user submits the payment information.

9 - The system shall allow the user to update its profile information.

10 – The system shall provide and administration portal.

11 – The system shall allow the creation of additional administrators.

12 – The system shall allow the administrator to view/edit/delete all registered users.

13 – The system shall allow the administrator to view/edit/delete the car inventory.

14 – The system shall allow the administrator to generate a sales report in a valid date range.

## 3.3 Nonfunctional requirements

(Johann)

## 3.4 Constraints (“Pseudo requirements”)

(Johann)

## 3.5 System models

(What should go here?)

## 3.5.1 Scenarios

(What should go here?)

## 3.5.2 Use case model

Use Case 1

* Name: Login
* Actor: User/Admin
* Entry Condition:
  + User/Admin must be at the login web page of Sunshine Online Car Dealer System.
  + User/Admin has a valid username and password.
* Event Flow:
  + User/Admin input username and password into the Login web page.
  + Sunshine Online Car Dealer System match username/password entered with list of stored users/admin.
  + Sunshine Online Car Dealer System logs User/Admin into the system.
  + User/Admin are redirected to home/index web page or into the requested page showing a logged in status.
* Exit Condition: User/Admin is logged into Sunshine Online Car Dealer System.
* Exceptions:
  + Cancel
  + “Wrong Username/Password”
  + “No connectivity”

Use Case 2

* Name: Purchase A Car
* Actor: User
* Entry Condition:
  + User must be at a web page of Sunshine Online Car Dealer System displaying a list of cars.
  + User must be logged in.
* Event Flow:
  + User selects car to be purchased.
  + Sunshine Online Car Dealer System displays the car details including price.
  + User selects Purchase option.
  + User enters the credit card information to purchase the selected car.
  + Sunshine Online Car Dealer System shows a confirmation web page with transaction details and delivery date.
* Exit Condition: User purchased a car.
* Exceptions:
  + Cancel
  + Invalid Credit Card Information
  + Sold to another faster customer

Use Case 3

* Name: Edit Cars
* Actor: Admin
* Entry Condition:
  + Logged in as admin
  + On the page with the “Edit Cars” function
* Event Flow:
  + Admin choose to do “Edit Cars”
  + Edit the information of the cars in the inventory
  + Submit to update the car’s information
* Exit Condition: Admin updated car information.
* Exceptions:
  + Cancel
  + Repeated Unique values

Use Case 4

* Name: Add User
* Actor: Admin
* Entry Condition:
  + Logged in as admin
  + On the page with the “Add User” function
* Event Flow:
  + Admin choose to do the “Add User”
  + Enter the information of the new user that needs to be added
  + Submit it to create the new user
* Exit Condition: A new user is created in the system.
* Exceptions:
  + Cancel
  + “Username already exists”

Use Case 5

* Name: Edit User Profile
* Actor: Admin
* Entry Condition:
  + Logged in as admin
  + On the page with the “Edit User Profile” function
* Event Flow:
  + Admin choose to edita User’s Profile
  + Edit the information of the selected User
  + Submit to update the user’s information
* Exit Condition: Admin updated user’s profile information.
* Exceptions:
  + Cancel
  + Repeated Unique values

Use Case 6

* Name: Add A Car
* Actor: Admin
* Entry Condition:
  + Logged in as admin
  + On the page with the “Add A Car” function
* Event Flow:
  + Admin selects the “Add A Car”function
  + Enter the information of the new car that needs to be added
  + Submit it to create the new car
* Exit Condition: A new car is added in the system.
* Exceptions:
  + Cancel
  + Repeated Unique values

Use Case 7

* Name: RemoveA Car
* Actor: Admin
* Entry Condition:
  + Logged in as admin
  + On the page of a list of cars with the “Remove” function
* Event Flow:
  + Admin selects the car needs to be removed
  + Select the “remove” function
  + System asks to confirm to do the remove
  + Admin confirm to remove the car
* Exit Condition: The selected car is removedfrom the system.
* Exceptions:
  + Cancel

Use Case 8

* Name: RemoveA User
* Actor: Admin
* Entry Condition:
  + Logged in as admin
  + On the page of a list of users with the “Remove” function
* Event Flow:
  + Admin selects the user needs to be removed
  + Select the “remove” function
  + System asks to confirm to do the remove
  + Admin confirm to remove the user
* Exit Condition: The selected user is removedfrom the system.
* Exceptions:
  + Cancel

Use Case 9

* Name: Search forA Car
* Actor: Guest/User
* Entry Condition:
  + On the home page of Sunshine Online Car Dealer System
* Event Flow:
  + Guest/User select desired car from the detail fields
  + Guest/User submit to search
  + Sunshine Online Car Dealer System displays a list of cars
* Exit Condition: Guest/User has a page displaying a list of cars.
* Exceptions:
  + “No cars found.”

Use Case 10

* Name: Register
* Actor: User
* Entry Condition:
  + User must be at the registration web page of Sunshine Online Car Dealer System.
  + User has anemail account and residence address.
* Event Flow:
  + User input personal information (First name, last name, email, username, password, billing address, shipping address).
  + User submits the information to register.
  + Sunshine Online Car Dealer System checks if username or email is not taken, and creates a new user account.
  + User is redirected to home page showing a logged in status.
* Exit Condition: User has an account in Sunshine Online Car Dealer System.
* Exceptions:
  + Cancel

Use Case 11

* Name: Update own Profile
* Actor: User/Admin
* Entry Condition:
  + User/Admin must be logged in.
  + User/Admin must be on customer profile web page of Sunshine Online Car Dealer System.
* Event Flow:
  + User/Admin selects update account information option.
  + User/Admin edit the personal information desired to update.
  + Submit to update its own information.
  + User/Admin is redirected to customer profile web page showing a logged in status.
* Exit Condition: User/Admin has updated its own profile.
* Exceptions:
  + Cancel
  + Repeated Unique values

Use Case 12

* Name: Generate Sales Report
* Actor: Admin
* Entry Condition:
  + Logged in as admin
  + On the admin home page with the “Generate Sales Report” function
* Event Flow:
  + Admin selects the date range of the desired report.
  + Select the “generate” function
  + System displays the sales report in the date range.
* Exit Condition: The system displays the sales report.
* Exceptions:
  + Cancel

## 3.5.3 Object model

(Roger)

Object Models are object-related diagrams of pages and pages of source code, and represent an overall organization of object types and their relationships. Although they don’t have a fine level of detail and can lack precision, they are easy to understand and provide a more appropriate “big picture” of the solution.

The purpose of Object Modeling is to represent objects from the “client world” as programming classes and objects. Even though these objects have a name similar to a concept from the problem domain, classes and objects are defined in their source code to behave and represent a related object from the “real world” that is needed in the system. Moreover, Object Models describes how this classes will work together to supply the client desired functionality by the type of link between them.

The Sunshine Online Car System will contain the following classes:

1. **Online Car System**
2. **User**
3. **Admin User**
4. **Regular User**
5. **Inventory**
6. **Car**
7. **Picture**
8. **Credit Card**
9. **Transaction**

**Online Car System Class**

The **Online Car System** class will be our system principal class. It will be responsible for providing searches to users and guest or allow guests to register in the scheme. The **Online Car System** class is the entry point to the car dealer since it will handle authentication, either login in or out of the system.

The principal attributes are **Inventory** and **User**. The **Online Car System** class will contain one **Inventory** object (we will discuss this object further on) and many **User** objects. 

**User Class**

The **User** class will model general user objects such as: Name, Middle and Last Name, Shipping and Billing Address, Username, Password, Email and Log Status. This class will serve as a super class to the **Admin User** and **Regular User** class.

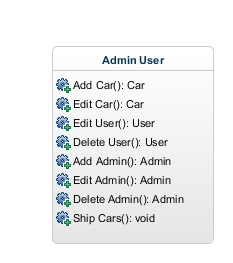


**Admin User**

The Admin User class inherits from the User class; however, its functionalities are different. It is in this class scope to:

* Add/edit either a Car or a Regular User object
* Add/edit/remove another Admin User objects.

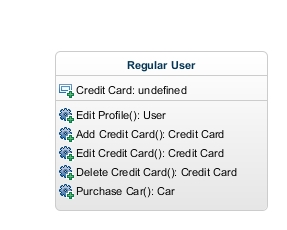
Another Admin User responsibility is to confirm or ship cars from inventory.



**Regular User**

The Regular User class will be distinctive because it will contain an attribute of class Credit Card. It will be constrained that at least one Credit Card is associated with the Regular User, but there will be no maximum in the quantity of this object for the Regular User class. The User will have full access and control over his/her Credit Card objects, being able to add, edit or remove them at any time.

The Regular User profile will consist in those attributes inherited from the User class except for the username. The Regular User profile will another feature left for the user to update. Regular User can purchase Car with Credit Card and there is no limit about the maximum Car objects a regular User can purchased. The more the better will be for the business.



**Inventory Class**

The Inventory class will behave as car warehouse; hence, will contain attributes of type Car objects. Also, for programming benefits it will contain an integer field with the available quantity of vehicles for sale.



The relation between class Inventory and Car will be from one to many. The Inventory class allows many Cars object and again there is no limit on the maximum number of Car object within the inventory class.

**Car Class**

The Car class has attributes as follows: VIN (Vehicle Identification Number), Make-Model, Year, Price, Mileage, Color and Color Details, Condition and Comments, Pictures and Car Status. The VIN number will be unique for each car, Price value can’t be zero or negative and the Comments attribute will contain any specific luxury or especial feature within the car. This class will include a maximum of five pictures represented by class Picture. Finally, the Car Status attribute will provide the current status of the Car class (e.g. SOLD, AVAILABE).



**Picture Class**

The functionality of this class is to model an image of the Car. This will be done by providing the address in hard drive where this picture is located.



**Credit Card Class**

The Credit Card class attributes are the name of the person in that Credit Card class, the Credit Card number, expiration date, security code and the type. This class will always be an attribute of the Regular User class and never instantiated from other classes. 

**Transaction Class**

The responsibility of the transaction class is to define the type of transaction. So far in this implementation we only purchase cars but other transaction types can be implemented here.



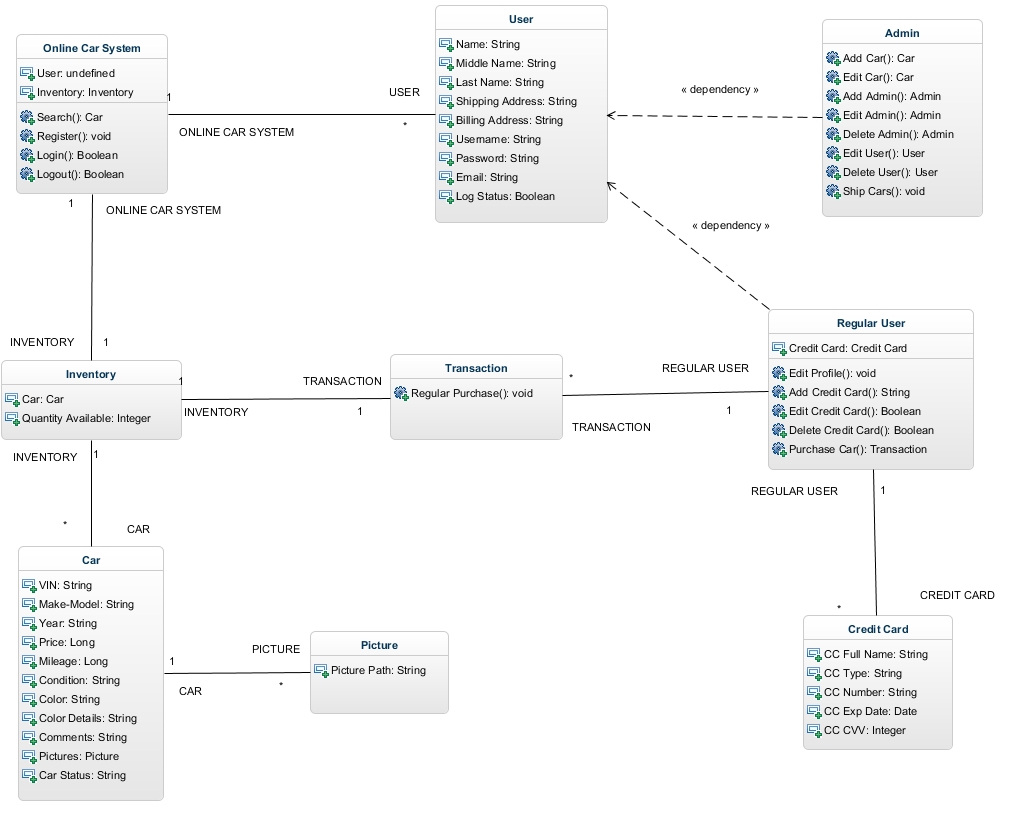
The Regular Purchase method is to be called by a Regular User class and defined to sale one object of type Car to the user. It is expected to estimate time of delivery and to change the Car Status property in the Car class, as well as charging the price of the object Car into the Regular User Credit Card object.

### 3.5.3.1 Data dictionary

(Roger)

### 3.5.3.2 Class diagrams

(Roger)



### 3.5.3.1 Data dictionary

(Roger)

### 3.5.3.2 Class diagrams

(Roger)

## 3.5.4 Dynamic models

(Sequence Diagrams)

(Ariel / Johann)

## 3.5.5 User interface

(Roger)

# 4. Glossary

Yang (Terminology)