

# **Report 1**

**Course:**

CSCI 441 VA Software Engineering

**Project Title:**

A Web Based Application for Vehicle Sales, Purchase and Inventory Management

**URL of Project Website:**

<https://web-app-for-vehicle-sales-dev.herokuapp.com/>

**Submission Date:**

02/23/2021

**Team-Members:**

Brady Neumann

Fatih Gurbuz

Ryan Shepard

Seth Whitaker

Taylor Cook

## **Breakdown of Contributions**

All team members contributed equally. As such, each member has typed their own name below.

Taylor Cook  
Brady Neumann  
Fatih Gurbuz  
Ryan Shepard  
Seth Whitaker

# **Table of Contents**

Cover Page: 1  
Breakdown of Contributions: 2  
Table of Contents: 3  
Customer Problem Statement: 4  
Decomposition into Sub-Problems: 8  
Business Goals: 10  
Enumerated Functional Requirements: 13  
Enumerated Nonfunctional Requirements: 14  
User Interface Requirements: 15  
Use Cases: 22  
User Interface Specification: 34  
System Architecture: 49  
Project Size (Use Case Points): 51  
Project Management: 55  
References: 56

# **Customer Problem Statement**

Car dealerships have a continually growing problem trying to keep up with today's technology. Nowadays, "70% of shoppers expected to find the ability to configure a payment on a dealership website, and 83% indicated that online buying technology would help them narrow down their vehicle choice and determine what is affordable" (V12, no date). Furthermore, with the prevalence of new car-buying websites such as Carvana, car shoppers do not need to set foot on a car lot in order to research, compare, and ultimately buy a car.

From the perspective of the consumer, one of the most prevalent problems with any type of shopping is being able to compare products. This problem exists not only for vehicles but for all electronic-styled purchases. The wide array of choices available, combined with limited availability for comparison, especially between brands, leaves the consumer feeling overwhelmed. Vehicles are one of the most problematic when it comes to this overwhelming feeling. Choices seem endless, comparing vehicles is challenging due to large amounts of variables, and the costs are immensely high. These types of issues are what has allowed online-only car selling to boom in the last couple of years.

Every dealership is encountering these sorts of problems in one way or another; some may have already solved the problem, but others are struggling. This is where a product needs to fill the gap. Dealers need a solid online presence where customers feel that they can shop without buyer's remorse. Whether this comes in the form of searching for vehicles then test driving them at the dealership, or purchasing completely online, the customer needs to be able to dictate their own buying experience.

The customer should be able to sort through vehicles based on multiple different criteria (price, brand, consumer rating, make, model, color, etc.). This ability for comparison will allow the customer to feel confident in the car they are choosing. This, coupled with a very easy to navigate site, makes the car-buying experience a one-stop-shop.

The dealers themselves face data problems. Keeping track of sales data as well as inventory management is a challenging task. If done incorrectly, a misplaced file, or mistyped keystroke can lead to wildly inaccurate information. The application needs to keep track of all of this data for the dealer. Historic sales data should be kept safe for reference whenever the dealer wants to access it, and only admin login credentials should be able to edit these data points as needed. Employee logins should allow use of the data without the ability to create, edit, or delete, and customers or users who have not logged in will not be able to access anything. Inventory management needs to be in a similar state. Admins can add, delete, or edit prices or availability, while employees and customers will only be able to see the data. Another major feature needs to be the ability for the admin to edit the data through an interface rather than requiring any coding. Thus, someone with no technical knowledge at all will still be able to oversee the site. In short, the benefit of all of this is automating a series of tasks so that it removes the human element as much as possible. The security should allow for the database to be protected from anyone other than authorized users, and historic sales data will be kept automatically.

Another issue facing dealers is the cost of the website. Websites are often expensive and getting up off the ground usually has large one-time costs associated with it. The ability to scale is key, as it is incredibly important that should the site need to grow, it does not need to be remade. Upkeep requirements should be low, as no one on staff will know how to fix problems

that arise. Instead, everything should be able to be managed by someone with no technical ability interacting with various interfaces. In the future, additional features may need to be added.

Security will also need to be one of the primary concerns. If unauthorized people are able to change prices or listings, it could cost enough money to put a dealership out of business. Similarly, a normal employee should not be able to manipulate data as this is both a security and error-proofing requirement. An admin, or multiple admins, should be able to log into the site and make changes, but no one else should have access to any sensitive information. No price changes, inventory adjustments, or any editing should be able to be performed by anyone other than the approved admins. Furthermore, some data should only be available to employees and not to customers. Customers should not be able to see historic sales data in any form, nor should they be able to see inventory amounts as they could use either of these to gain a leg up in price negotiations. Finally, it is incredibly important that malicious attackers be unable to take the website down within reason. A dealer will not have the money for security like a larger company would, but an individual who is looking to take down the site for fun or for profit should be unable to.

Many sites scare customers away by having either too much information or not letting them access the content until they sign up. After the customer does sign up, sites often just use the customer's information to target them with ads, emails, and incessant car-buying information. The web application needs to serve the customer a pleasant experience without shoving ads down their throat or trying to get them to sign up instantly. The site should allow a customer to browse as they see fit and allow them to sign up and sign in for saved searches and preferences. Customers need to find it easy to navigate the pages and information, and if they do wish to purchase something, the process should be as simple as possible.

Car information should be shown in a simple layout on some sort of virtual card, that when clicked, takes the customer to a more detailed layout of that car's information. The "cards" should contain brief descriptions, key information such as price, and an image of the vehicle. The detailed information should have everything a customer might want to know about a car and possibly a link to the manufacturer's website. In addition to cars, different car parts and accessories should be available for browsing and purchasing. All of these parts and accessories should be modeled in a similar way to the cars but in a different section of the site.

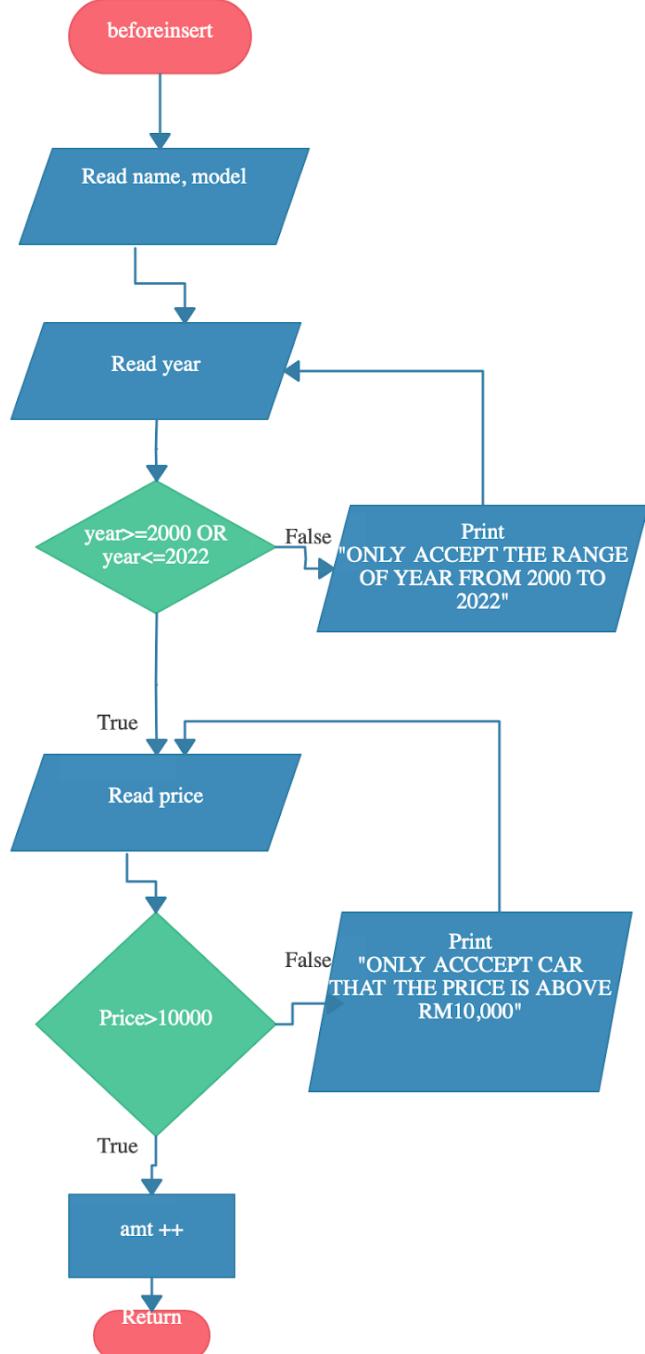
Customers should be able to add multiple cars and parts to a cart, which will be saved in case the customer leaves the page. Once the customer has added at least one item to the cart, they should be able to click a checkout button, which will take them to a checkout page. This checkout page should provide payment options, financing estimates and dealership location for vehicles, and forms to fill out necessary information. The page should provide a submit button that the customer can click when they have finished filling out the information. This page should also verify that all necessary information is filled out. If not, the page should highlight missing information, and prompt the customer to complete it. If all information is complete, the application should display a confirmation page with a receipt.

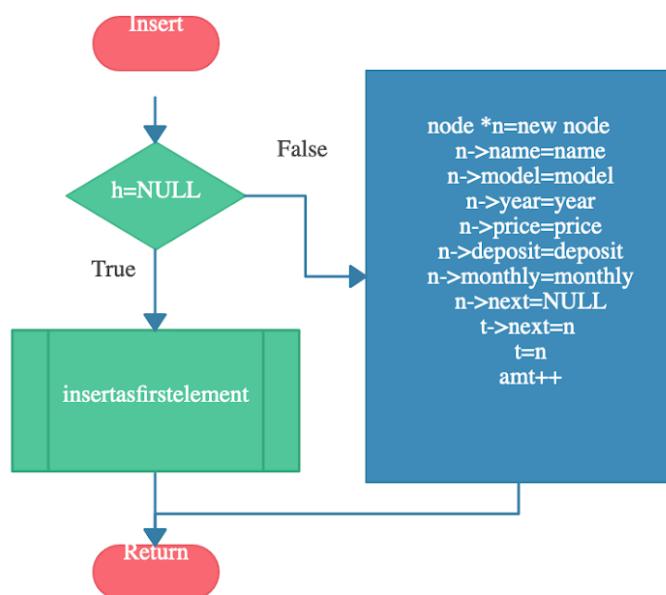
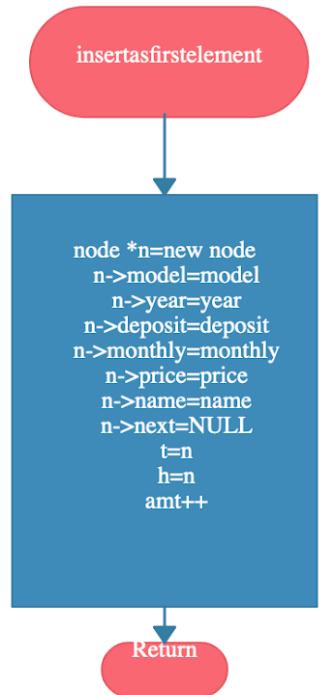
# Decomposition into Sub-Problems

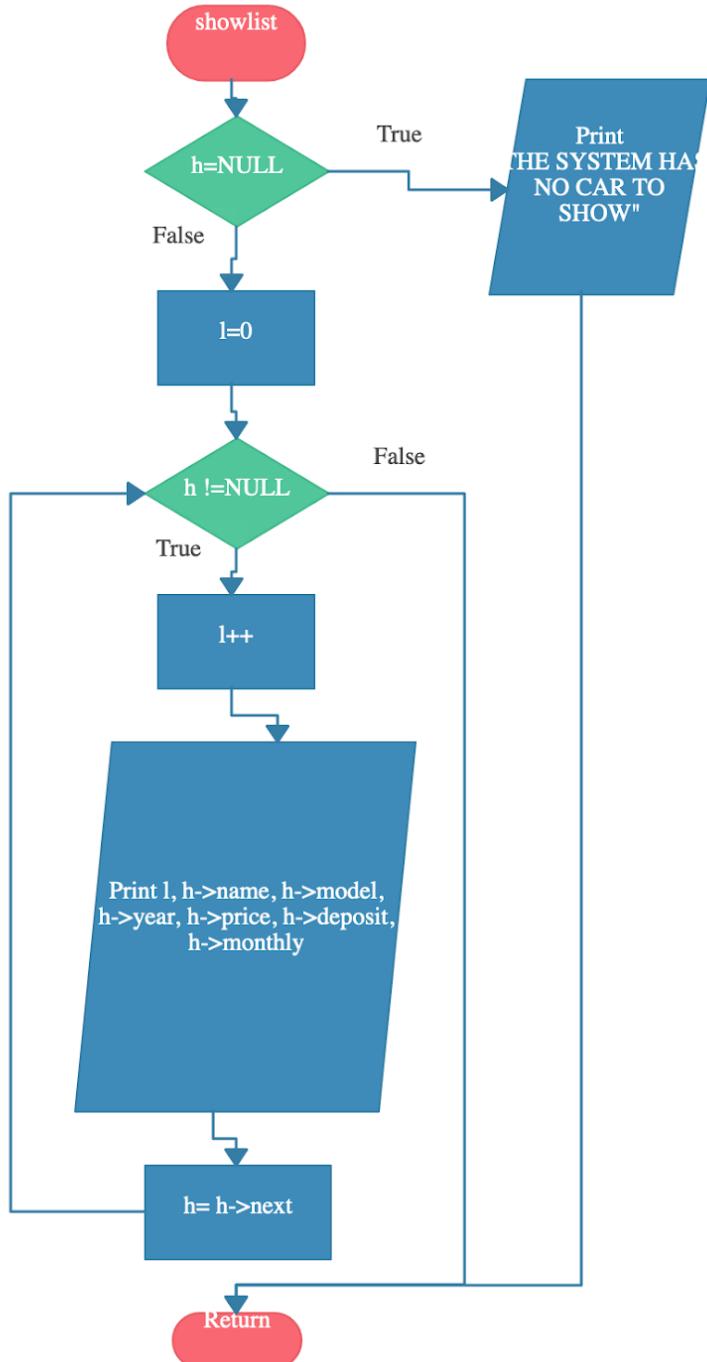
- Customer Needs
  - Simplicity
    - Comparisons
      - Browsing by brands/color/model/etc.
      - Sort options
        - Database with all related info
    - Search Bar
      - Being able to query everything available at once
        - Database is searchable through this search bar
      - Interface for search bar
        - Front end work (Possibly ReactJS)
    - Comparison tool
      - Front end can show multiple options side by side
  - Online-Only
    - Ability to purchase and research without employee interaction
      - Email form if questions are needed
      - Straightforward options that anyone could understand
  - Navigation
    - User sign in allows for saved queries
      - Ability to save previous searches
        - Only if the user has a sign in
    - Visually appealing and easy to use
      - Design work
      - Simple layout
      - Navigation bar is constant throughout website
  - Low Pressure
    - Independence
      - No Required Sign Up
        - Browsing without sign in
      - Fully functional from start to finish
        - Cart
        - Checkout
        - Purchase Options
          - Options available on each car/part

- No invasive ads
  - If ads are included, they are sidebar only and do not error out
  - No auto playing ads or sounds
- No data-mining/lead generation
- Dealership Needs
  - Security
    - Protection from malicious activities
      - Verification of data
        - Filtering all user input taken into site
      - Password hashing system
        - Secured database of user logins
    - Separation of permissions
      - Ability for front end to show different options depending on user-type
        - Back end needs to serve up info depending on login
      - Customer
        - Not able to see historic sales data
        - Not able to see inventory amounts
      - Employee
        - Not able to edit/create/delete historic sales data
        - Not able to edit/create/delete inventory amounts
      - Admin
        - Ability to edit/create/delete historic sales data without coding
        - Ability to edit/create/delete inventory amounts without coding
        - Interface specifically for admins
  - Cost
    - Possible future expansion
      - Allow scalability
        - Well used comments throughout
        - Ability to scale database (MySQL Community Edition)
    - Low cost
      - Work on needs first, then wants
        - Completion of base site, then add in extras
    - As small of database as possible
      - Fits within MySQL Community

# Business Goals







# Enumerated Functional Requirements

Identifier	Priority	Requirement
REQ-1	1	The application should allow the user to sort through vehicles based on multiple different criteria (price, brand, consumer rating, make, model, color, etc.).
REQ-2	5	The application should keep track of inventory and sales history, with sales history data being tracked automatically.
REQ-3	4	The application should provide 3 levels of authentication: Customer, Employee, and Admin.
REQ-4	3	The application should allow Admins to view, create, edit, and delete all data.
REQ-5	2	The application should allow Employees to view inventory and sales history data without the ability to create, edit, or delete it.
REQ-6	2	The application should allow Customers to view inventory data, without the ability to create, edit, or delete it, however, they should not be allowed to view inventory numbers.
REQ-7	4	The application should provide a graphical interface for Admins to view, create, edit, and delete data without the need for any coding.
REQ-8	3	The application should display inventory information in a simple layout on a virtual card. These cards should contain brief descriptions, key information, and an image of the vehicle.
REQ-9	4	The application should display a page with detailed information when the corresponding card is clicked that contains all the information a customer might want to know.
REQ-10	3	The application should allow Customers to add multiple cars and parts to a cart. The cart should be saved to allow Customers to leave the page and come back. A checkout button should be displayed after at least one item has been added to the cart, which will take the customer to a checkout page.
REQ-11	1	The checkout page should provide financing estimates, dealership location selector, payment options, and forms for necessary information.
REQ-12	2	The checkout page should verify all necessary information is filled out, highlighting all missing information on submitting. If all necessary information is filled out on submitting, a confirmation page with a receipt should be loaded.

# Enumerated Nonfunctional Requirements

<b>Identifier</b>	<b>Priority</b>	<b>Requirement</b>
REQ-13	2	The user interface must be user-friendly and easy to use.
REQ-14	1	The user interface of the system must be simple enough so that any user can use it with a minimum training.
REQ-15	5	In case of failure, the system must be easy to recover and must suffer minimum loss of important data.
REQ-16	5	The application should be always readily available for concurrent use (by multiple users)
REQ-17	4	The application should be designed and work as a distributed system (running on multiple machines)
REQ-18	2	The application should have reusable design, including modularity in coding for future updates.
REQ-19	5	The application should have reliable features that enable data to be transmitted securely through the system.
REQ-20	4	The application should save errors faced by users and report them to admin.
REQ-21	3	The application should have a well defined theme that will be used throughout the whole website.
REQ-22	1	Each page in the website should be loaded relatively quickly.
REQ-23	4	The application should have a clear distinction between customer, employee, and admin account and make this distinction clear to users.
REQ-24	1	In case of a maintenance routine, the customers should be warned at least a day prior.
REQ-25	5	The application should be available at any time of the day

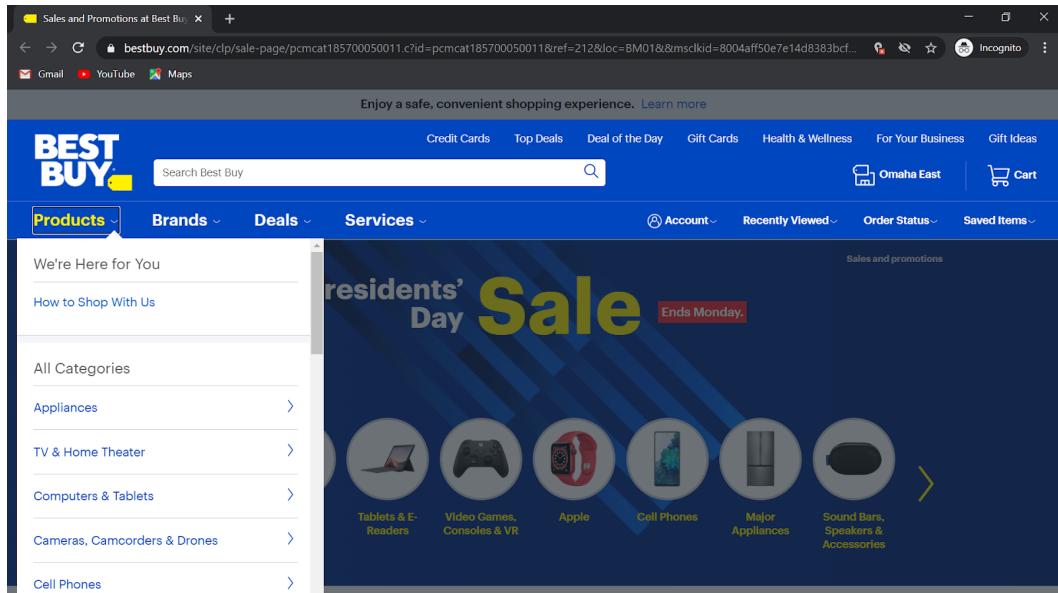
# User Interface Requirements

## I. Easy to navigate

A. **Description:** The user should not have any issues figuring out how to navigate the web application. Users should be able to find anything that they are looking for with navigation options such as a navigation menu, filter and sort options, and a search bar.

B. **Priority:** 2

C. **Graphic:**



(Includes many drop down navigation menus, sub-navigation menus, and a search bar for an easy to navigate experience.)

Source: <https://www.bestbuy.com>

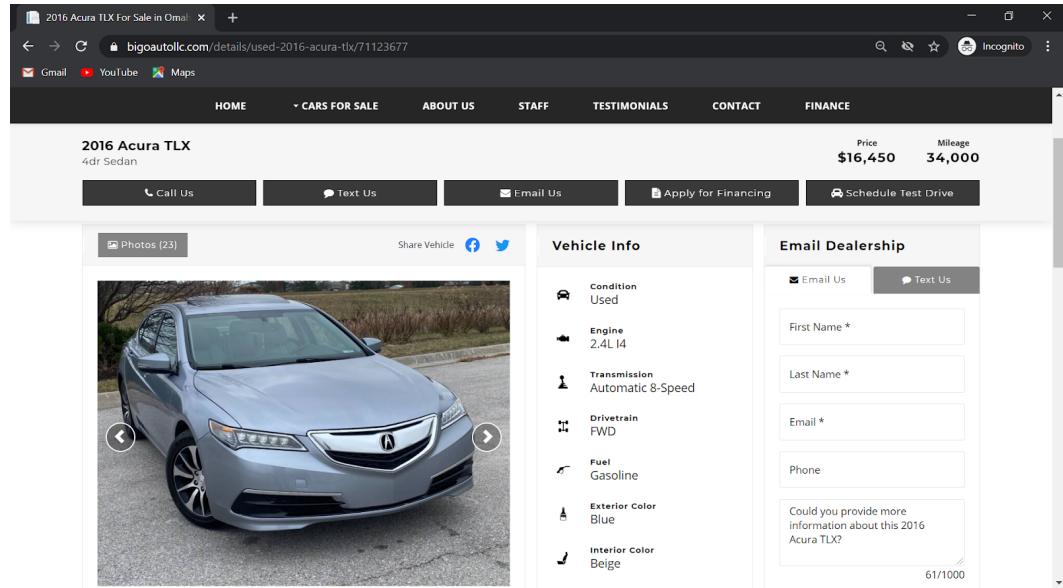
## II. Informative and concise

A. **Description:** The user interface should include everything that a user will need to know about a car that they are planning to buy. This information should be displayed in an easy to read format that will not overwhelm the user with

paragraphs of text per product.

### B. Priority: 3

### C. Graphic:



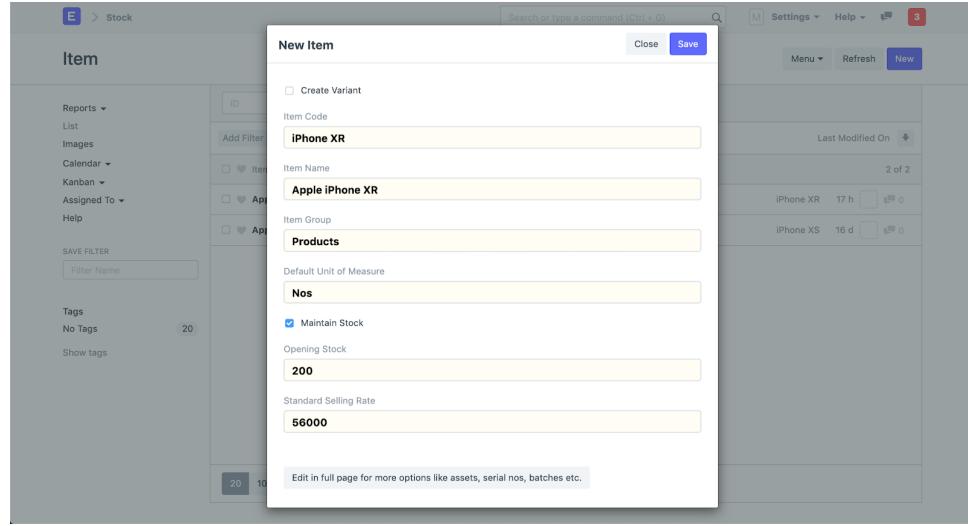
(Provides customers with price and mileage at the top, images on the left side, and concise information about the vehicle in the middle that is easy to process.)  
Source: <https://www.bigoautollc.com/details/used-2016-acura-tlx/71123677>

## III. Customizable

**A. Description:** The user interface should provide administrators with the ability to customize it without any coding knowledge. This may include adding, modifying, or deleting products.

### B. Priority: 4

### C. Graphic:



(Being able to add a new product easily without having to code it in.)

Source: [How to configure a Product Page](#)

#### IV. Scalable

**A. Description:** The user interface should be usable on devices with bigger screens

and smaller devices such as a phone.

**B. Priority:** 1

**C. Graphic:**



(Web application being usable for desktop and mobile devices.)

Source: [mock-up-desktop-and-mobile](#)

## V. Feedback

**A. Description:** The user interface should provide responsive feedback to the user when they interact with it. This may vary from reminding the user to enter any information in a form that they may have forgotten, to notifying them when they have added something to their cart.

**B. Priority: 2**

**C. Graphic:**

The screenshot shows a web browser window for Best Buy's checkout process. The URL is bestbuy.com/checkout/r/fulfillment. The page has a blue header with the Best Buy logo and 'Checkout' text. On the left, there's a 'Getting your order' section with 'Shipping Information' fields for First Name (Brady) and Last Name (highlighted in red with error message: 'Please enter a last name.'), Address (highlighted in red with error message: 'Please enter address.'), City, and State. To the right is an 'Order Summary' table:

Order Summary	
Shipping	Get It by Wed, Feb 17 ASUS - 14.0" Laptop Intel Celeron N4020 - 4GB \$249.99 Qty 1 Remove Send a gift message
Digital Delivery	Webroot Internet Security with Antivirus (3) FREE Qty 1 Remove
Item Subtotal \$249.99 Shipping FREE Estimated Sales Tax \$17.50 + Apply a Best Buy Tax Exempt Account Number	
Total	\$267.49

At the bottom, a note says 'All applicable sales tax, shipping, and delivery prices are estimates.' and a 'Help' button is visible.

(The website highlighted the form entries I still need to fill out and printed a message below each one before it will let me check out.)

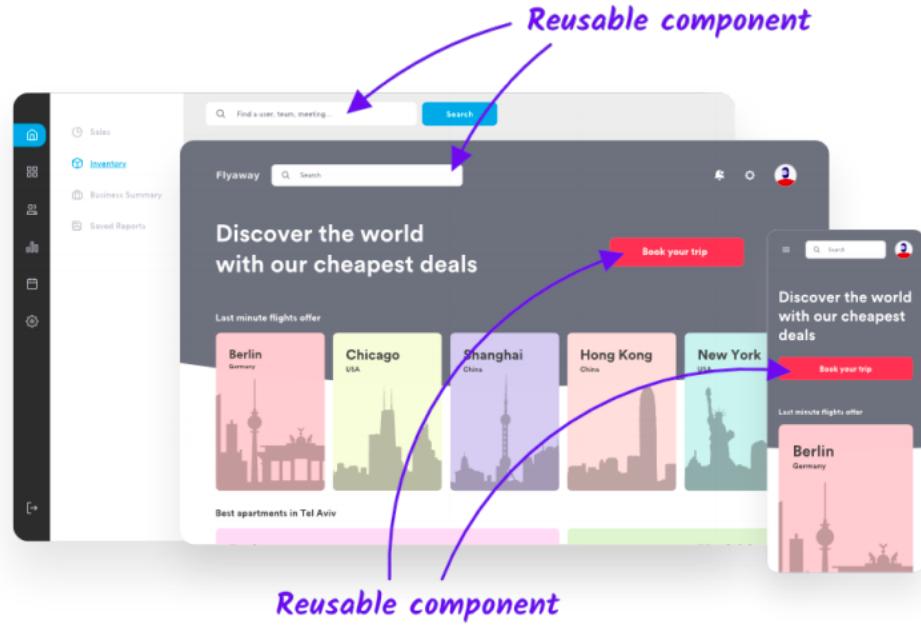
Source: <https://www.bestbuy.com>

## VI. Consistent

**A. Description:** The user interface should be consistent across all of the different pages. This may include the general layout, fonts used, buttons, and icons.

**B. Priority: 2**

**C. Graphic:**



(Shows reusing certain content to provide a consistent look across the web application.)

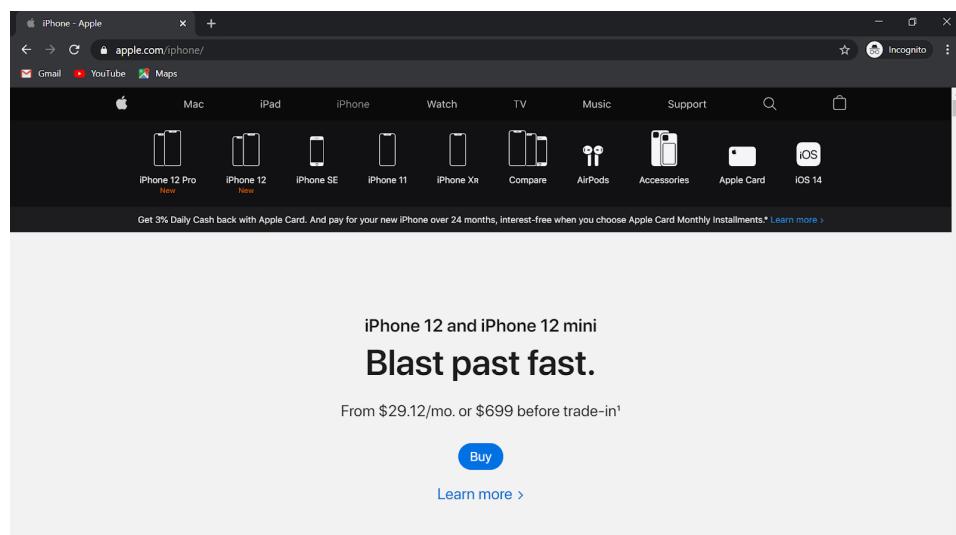
Source: <https://blog.bitsrc.io/5-ways-to-improve-ui-consistency-99013bf20417>

## VII. Attractive

**A. Description:** The user interface should be pleasant to look at and eye catching to the user.

**B. Priority: 4**

**C. Graphic:**



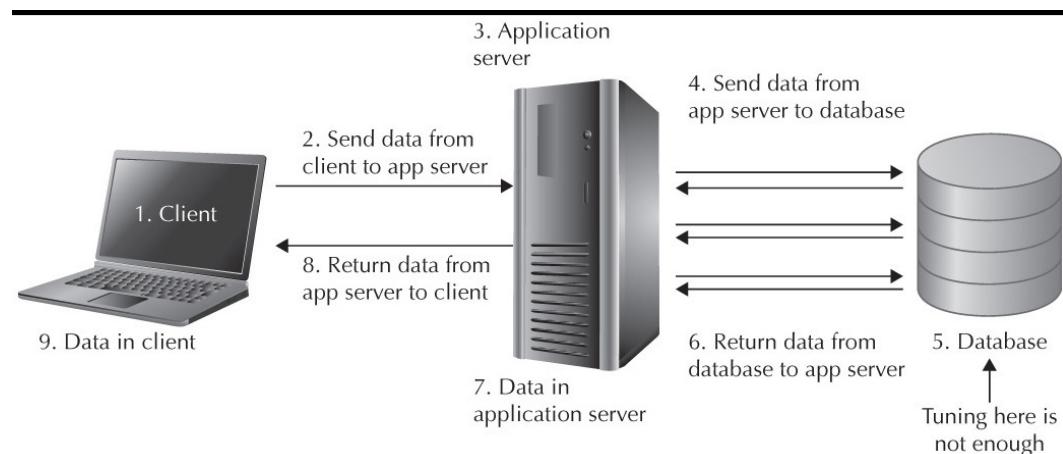
(Deciding to use small images of the products in the navigation menu is an example of grabbing more attention from the user rather than simple text.)  
 Source: <https://www.apple.com/iphone/>

## VIII. Efficient

**A. Description:** The user interface should process data quickly and not have the user waiting for it to respond.

**B. Priority:** 3

**C. Graphic:**



(Cycle for data processing in web application.)

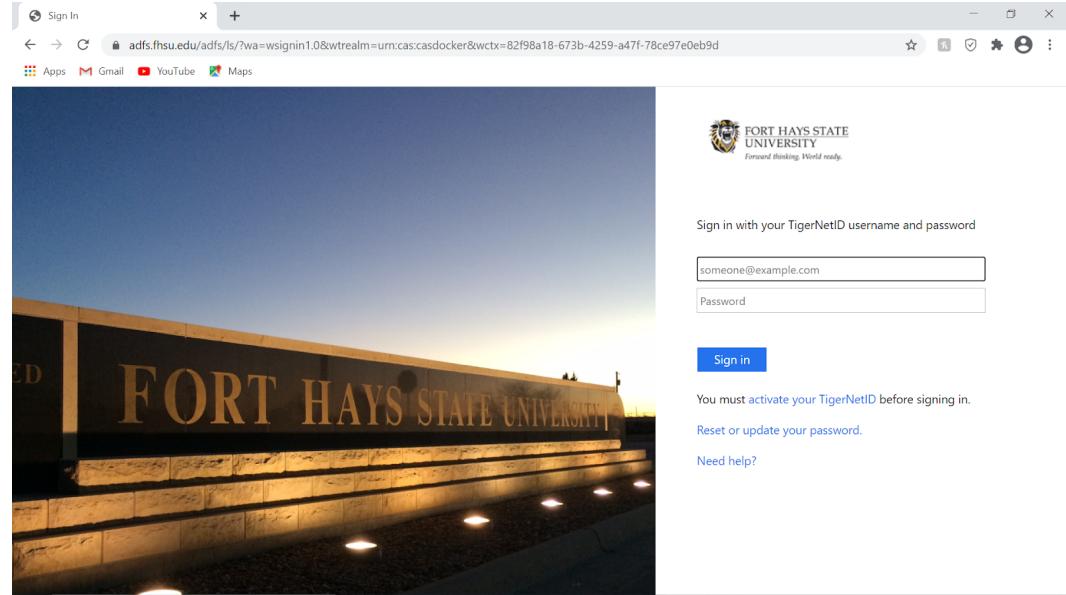
Source: <https://logicalread.com/application-flow-oracle-mc08/#.YCfXl2hKjb0>

## IX. Secure

**A. Description:** The user interface should ensure that data is secure and not allow a data breach to occur. This can include preventing a customer from editing the products by having an administrator login system, and protecting the financial information of the customer.

**B. Priority:** 4

**C. Graphic:**



(It is demanded that I login before I am able to access in order to prevent someone else from accessing my data.)

Source: <https://blackboard.fhsu.edu/>

# Use Cases

## Stakeholders

The stakeholders who hold an interest in this web application consist of customers, dealer employees, and dealer administrators.

## Actors and Goals

Actor	Actor's Goal	Type	Use Case Name
Customer	Sort by category	Initiating	Sort (UC-1)
	Search via search bar	Initiating	Search (UC-2)
	Compare options	Initiating	Compare (UC-3)
	Sign up	Initiating	SignUp (UC-4)
	View information about vehicles/parts	Initiating	ViewCarInfo (UC-5)
	Email contact form to dealer	Initiating	EmailDealer (UC-6)
	Add item to cart	Initiating	AddItem (UC-7)
	Remove item from cart	Initiating	RemoveItem (UC-8)
	Checkout	Initiating	Checkout (UC-9)
Employee	Read sales data	Initiating	CrudSalesData (UC-10)
	Read inventory data	Initiating	CrudInventoryData (UC-11)

Admin	Edit/create/delete sales data	Initiating	CrudSalesData (UC-10)
	Edit/create/delete inventory data	Initiating	CrudInventoryData (UC-11)
	Add employee	Initiating	AddEmployee (UC-12)
	Remove employee	Initiating	RemoveEmployee (UC-13)
Application	Route email form from customer to dealer email	Participating	EmailDealer (UC-6)
	Record/store sales data	Participating	Checkout (UC-9)
	Record/store inventory data	Participating	CrudInventoryData (UC-11)
	Display content	Participating	All Use Cases
	Error generation/tracking	Participating	All Use Cases
	Filter user data entered	Participating	UC-4, 6, 7, 8, 9, 10, 11, 12, 13
	Hash passwords	Participating	UC-4, 12
	Storage of user login data	Participating	UC-4, 12
	Authenticate user	Participating	Authenticate (UC-14)

## Casual Descriptions

<b>Use Case Name</b>	Sort (UC-1)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers can come to the site looking for a vehicle/part, and they would like to filter by the specific categories that interest them (e.g. brand, make, model, color, price, etc.). The application then filters by the customers selected criteria, then displays results.

<b>Use Case Name</b>	Search (UC-2)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers can come to the site looking for a vehicle/part, and they would like to search by certain criteria. A search bar allows them to enter any keyword they like, and the application filters and displays the results.

<b>Use Case Name</b>	Compare (UC-3)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers can come to the site looking for a vehicle/part, and they would like to compare different options. The customer should be able to select vehicles/parts to be compared side by side. The application then displays the content requested.

<b>Use Case Name</b>	SignUp (UC-4)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers can sign up to the application. The application will prompt the customer for an email/username and password. Upon receiving the data from the customer, the application filters and checks the data against already stored user login data. If the email/username is new, the application creates a new set of information in the user login datatable after hashing the password entered. If the email/username already exists, the application outputs to the customer that there is already a login with that information.

<b>Use Case Name</b>	ViewCarInfo (UC-5)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers will be able to see information about any vehicle/part they would like. They can go about this by viewing the “cards” available while sorting or viewing products. The customer can then click one of these cards to view more detailed information about the product. The application will serve up whatever is sorted/clicked on/available for the customer.

<b>Use Case Name</b>	EmailDealer (UC-6)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers who wish to email a content form to the dealer employees (this can be a question, request for contact, etc.), can fill out the form on the application and click submit to send it to the dealer. The application then verifies that this information is filled out correctly, and sends an email to the dealer.

<b>Use Case Name</b>	AddItem (UC-7)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers can add any available vehicle/part to their cart. The application will then store the item inside of the customer’s cart until they either checkout or delete the item from their cart.

<b>Use Case Name</b>	RemoveItem (UC-8)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers can delete any vehicle/part from their cart. The application will then remove the item from their cart.

<b>Use Case Name</b>	Checkout (UC-9)
<b>Actors</b>	Customer, Application
<b>Description</b>	Customers who wish to checkout with the items currently in their cart can enter their payment information and the application will process the payment. The application will then edit the sales data and inventory data accordingly. The application will then email the customer a receipt.

<b>Use Case Name</b>	CrudSalesData (UC-10)
<b>Actors</b>	Admin, Employee, Application
<b>Description</b>	An admin can choose to create, read, update, or delete sales data. An employee can only read sales data. The application will allow a specific interface for these options by employees/admin so that no coding is required. Any information that will be created or updated will be filtered to ensure the data fields are correct before submission. The application will then update the database according to changes made.

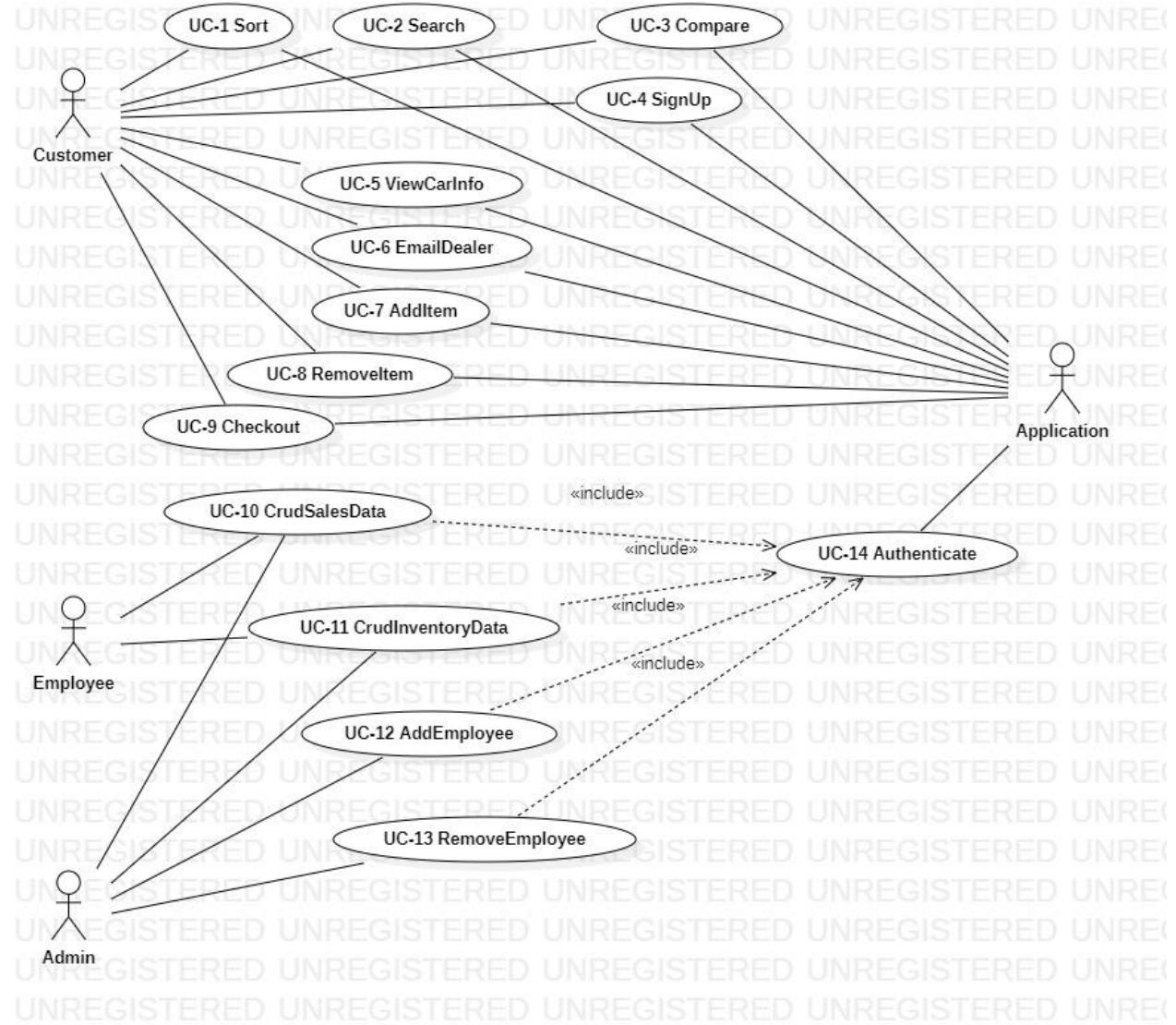
<b>Use Case Name</b>	CrudInventoryData (UC-11)
<b>Actors</b>	Admin, Employee, Application
<b>Description</b>	An admin can choose to create, read, update, or delete inventory data. An employee can only read inventory data. The application will allow a specific interface for these options by employees/admin so that no coding is required. Any information that will be created or updated will be filtered to ensure the data fields are correct before submission. The application will then update the database according to changes made.

<b>Use Case Name</b>	AddEmployee (UC-12)
<b>Actors</b>	Admin, Application
<b>Description</b>	An admin can choose to add an employee login to the application. The application will ensure that the username/email does not already exist in the user login database, then store the new employee login data to the database. This includes hashing the password before storage.

<b>Use Case Name</b>	RemoveEmployee (UC-13)
<b>Actors</b>	Admin, Application
<b>Description</b>	An admin can choose to remove an employee login to the application. The application will ensure that the username/email already exists within the database, then allow the admin to delete the login data. The application will then make the corresponding changes within the database.

<b>Use Case Name</b>	Authenticate (UC-14)
<b>Actors</b>	Application, Customer, Employee, Admin
<b>Description</b>	The application will authenticate any user signing into the application from the user login database. Upon login, the application will check the entered data against stored data to ensure that the correct person is logging in.

## Use Case Diagram



## Traceability Matrix

Req	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14
REQ1	1	x	x			x									
REQ2	5									x	x	x			
REQ3	4			x		x						x	x	x	
REQ4	3										x	x			
REQ5	2										x	x			
REQ6	2	x	x			x									
REQ7	4										x	x			
REQ8	3	x	x			x		x	x						
REQ9	4				x			x	x						
REQ10	3		x					x	x	x					
REQ11	1									x					
REQ12	2									x					
Max PW	3	3	3	4	4	4	4	4	5	5	5	4	4	4	4
Total PW	6	6	3	4	10	4	10	10	11	14	14	4	4	4	4

## Fully-Dressed Description

**Use Case UC-5:**

**ViewCarInfo**

**Related Requirements:** REQ-1, REQ-6, REQ-8, REQ-9, REQ-13, REQ-14

**Initiating Actor:** Customer

**Actor's Goal:** View information about vehicles/parts

**Participating Actors:** Application

**Preconditions:** The customer is viewing the application on their browser

**Postconditions:** The application has served the information in whatever form the customer requested

**Flow of Events for Main Success Scenario:**

- 1. Customer selects option to view products (sort, search, card click, etc.)
- ← 2. The application takes input from user, and displays information accordingly

**Use Case UC-9:****Checkout**

**Related Requirements:** REQ-2, REQ-10, REQ-11, REQ-12

**Initiating Actor:** Customer

**Actor's Goal:** Checkout and arrange payment for products in cart

**Participating Actors:** Application

**Preconditions:**

- The customer has items in their cart
- The customer has clicked the checkout button

**Postconditions:** The customer has purchased their items, and has been sent a receipt

**Flow of Events for Main Success Scenario:**

- 1. Customer clicks checkout button
- ← 2. Application displays the checkout page and form
- 3. Customer fills out form (address, credit card info, shipping, etc.)
- ← 4. Application confirms information
- ← 4a. Application updates databases
- ← 4b. Application emails receipt to customer
- ← 4c. Application displays success to customer

**Flow of Events for Extensions (Alternate Scenarios):**

- 1. Customer clicks checkout button
- ← 2. Application displays the checkout page and form
- 3. Customer fills out form (address, credit card info, shipping, etc.)
- ← 4. Application denies information, informs customer of incorrect data, and sends them back to checkout page and form
- 5. Customer fills out form again (4-5 can loop)

- ← 6. Application confirms information
- ← 6a. Application updates databases
- ← 6b. Application emails receipt to customer
- ← 6c. Application displays success to customer

### **Use Case UC-10, 11:      CrudSalesData, CrudInventoryData**

**Related Requirements:** REQ-2, REQ-4, REQ-5, REQ-7

**Initiating Actor:** Admin, Employee

**Actor's Goal:** Create, read, update, delete sales data

**Participating Actors:** Application

**Preconditions:**

- Admin/employee is logged in
- Admin/employee is viewing application interface for CRUD changes

**Postconditions:** The application has either displayed the information requested, or updated the databases as requested

#### **Flow of Events for Main Success Scenario:**

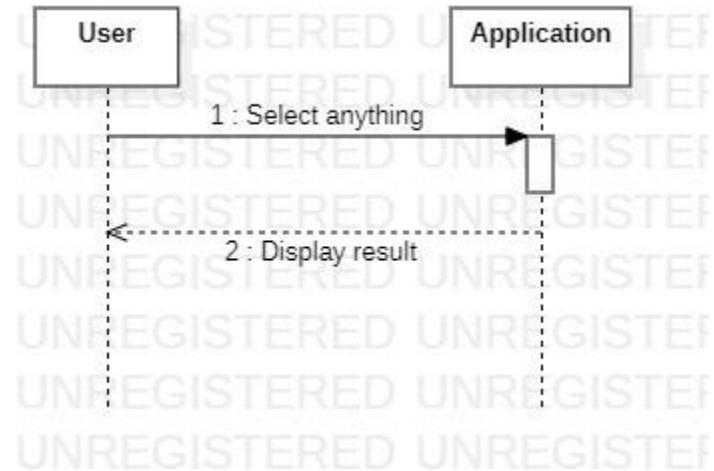
- 1. Admin requests change to information (or view in case of employee)
- ← 2. Application checks data, verifies it successfully
- ← 2a. Application updates databases
- ← 2b. Application outputs success to user

#### **Flow of Events for Extensions (Alternate Scenarios):**

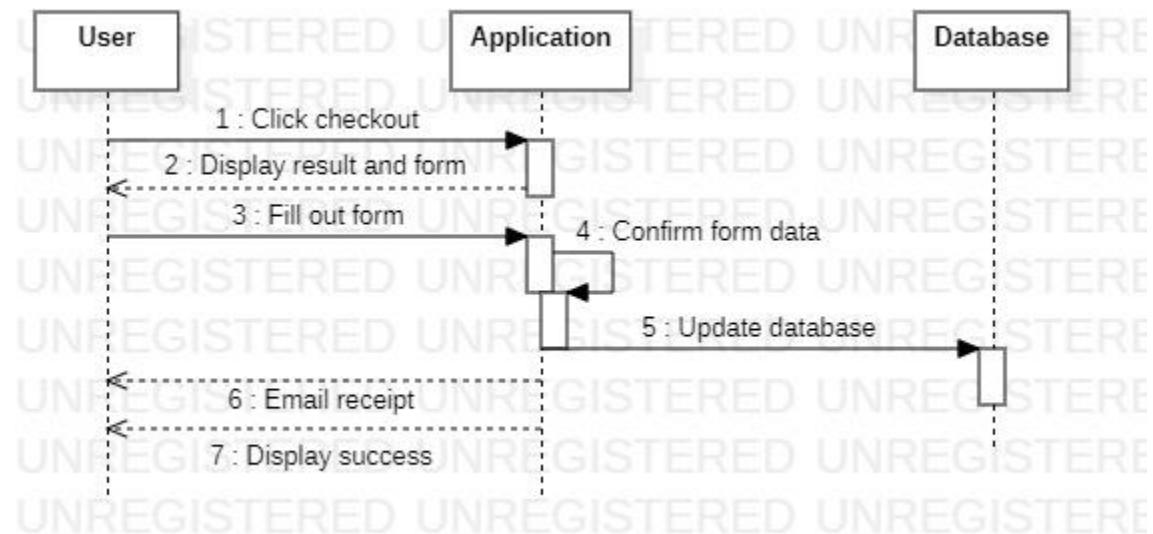
- 1. User request change to information
- ← 2. Application checks data, does not verify
- ← 2a. Application sends error message to user

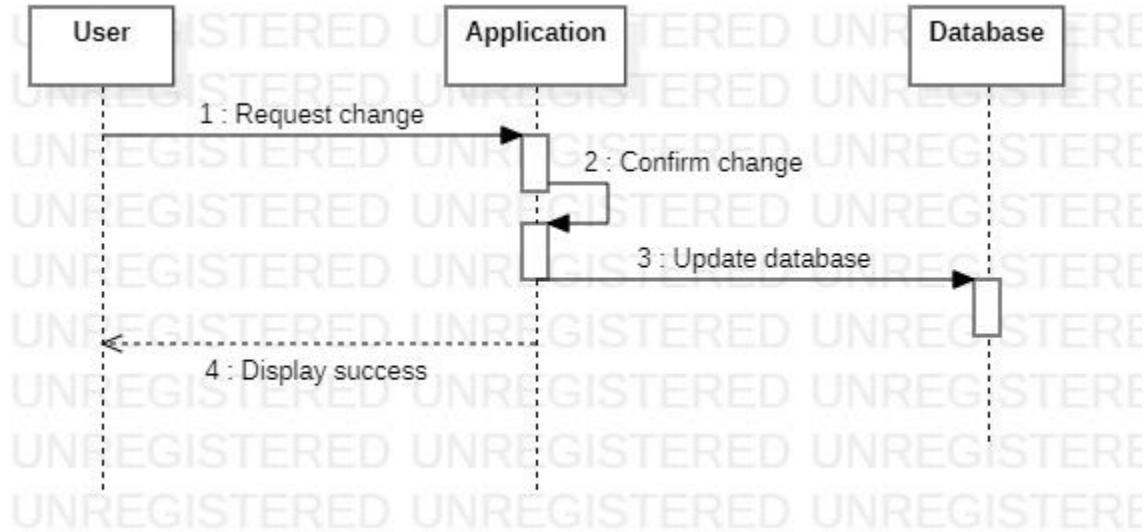
## System Sequence Diagrams

### ViewCarInfo (UC-5)



### Checkout (UC-9)



**CrudSalesData, CrudInventoryData (UC-10, UC-11)**

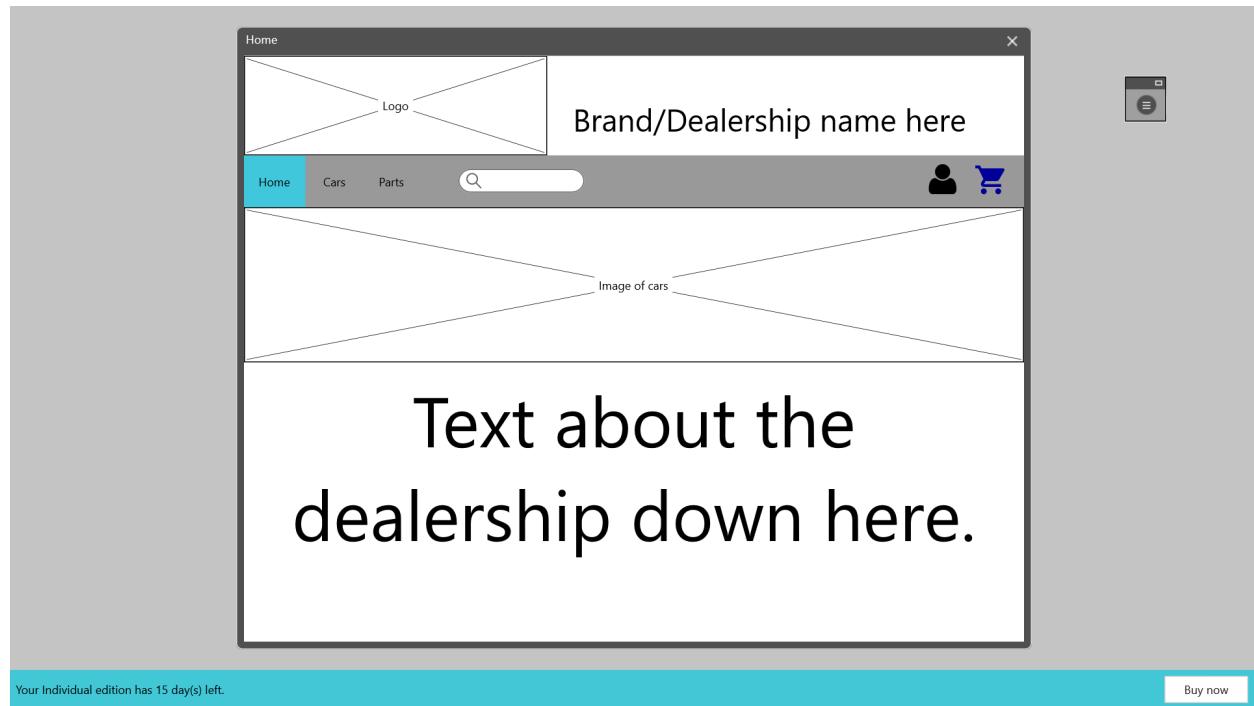
# User Interface Specification

The mockups in the pictures are a blueprint of the features that should be present in our web application and are not what the final product will look like. All mockups in this section of the report were created using the application MockPlus.

## I. Preliminary design

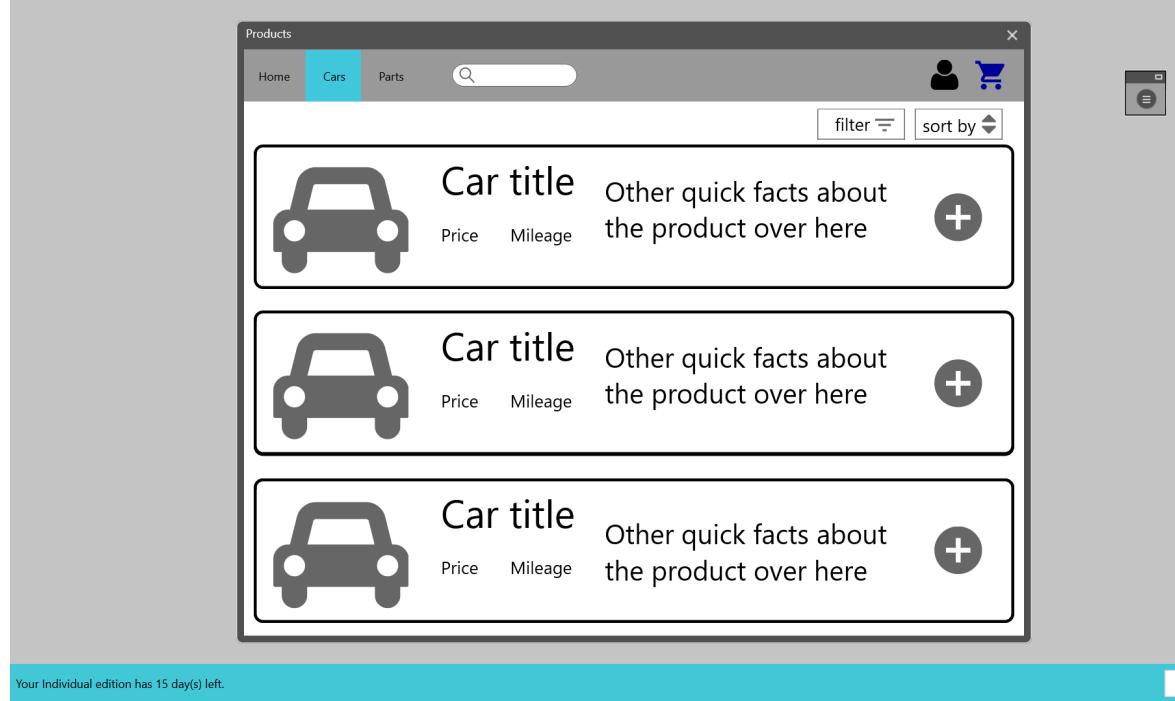
### A. UC-5: ViewCarInfo

Once the user enters the application, they will be greeted with the home page.

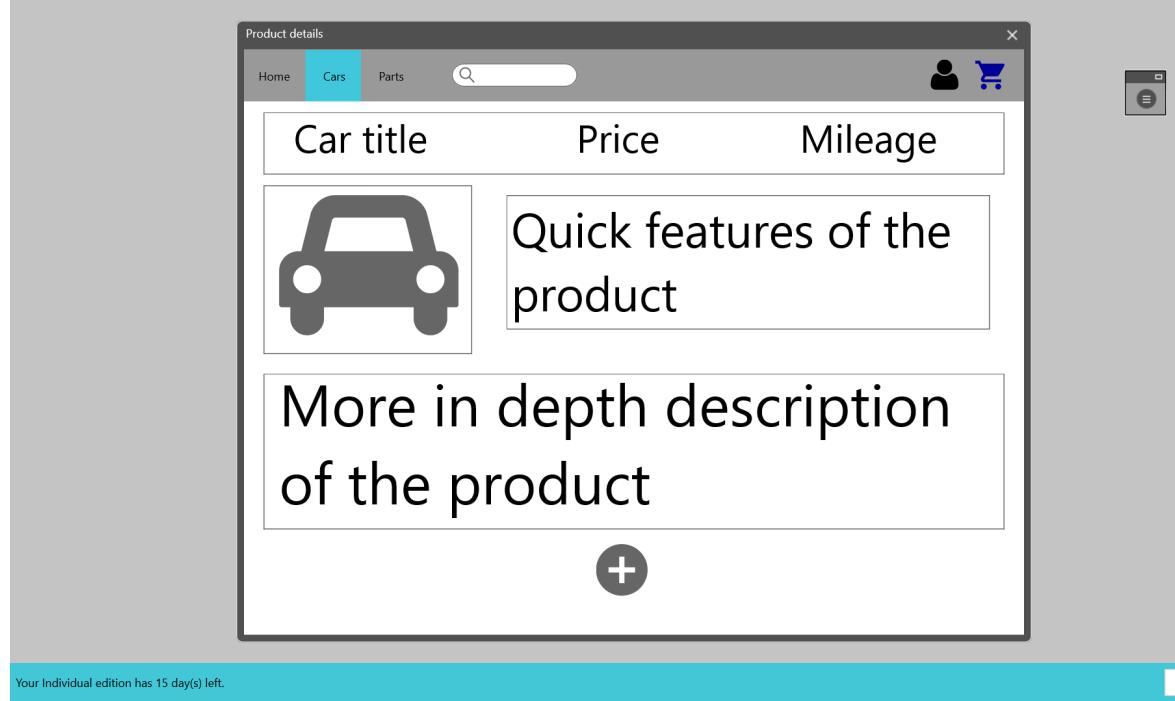


To view the cars that we have available, they can click on the ‘Cars’ tab at the top of the page. If they know exactly what they are looking for, then they can use the search bar that is also located at the top of the page to search for their desired car.

Once the user clicks on the ‘Cars’ tab, they will be taken to a new page with all of the available cars.



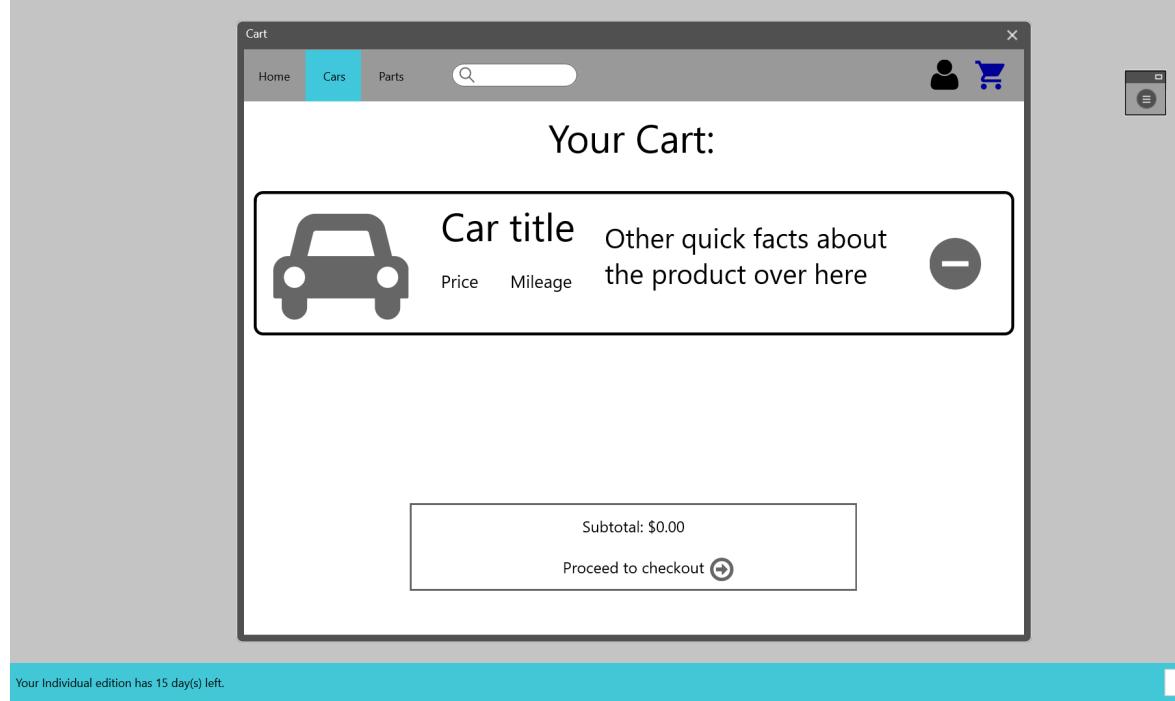
All of the cars we have available will be able to be seen in a set order. The user can change the order that the vehicles are displayed by clicking on the sort by button. They can choose to sort by price, mileage, and alphabetical, all in ascending and descending order. If the customer wants to limit the number of cars they see, they can use the filter button above the products. They can choose to filter products depending on make, model, price, or mileage. Once they have found a product they are interested in, they can add it to their cart by clicking the plus button on the card, or click on the card itself to get a more detailed view of the product.



This page will display all the information that a user needs to know about a particular vehicle that they are interested in. If they decide that they want to purchase the car, they can add it to their cart by clicking the button at the bottom of the page.

## B. UC-9: Checkout

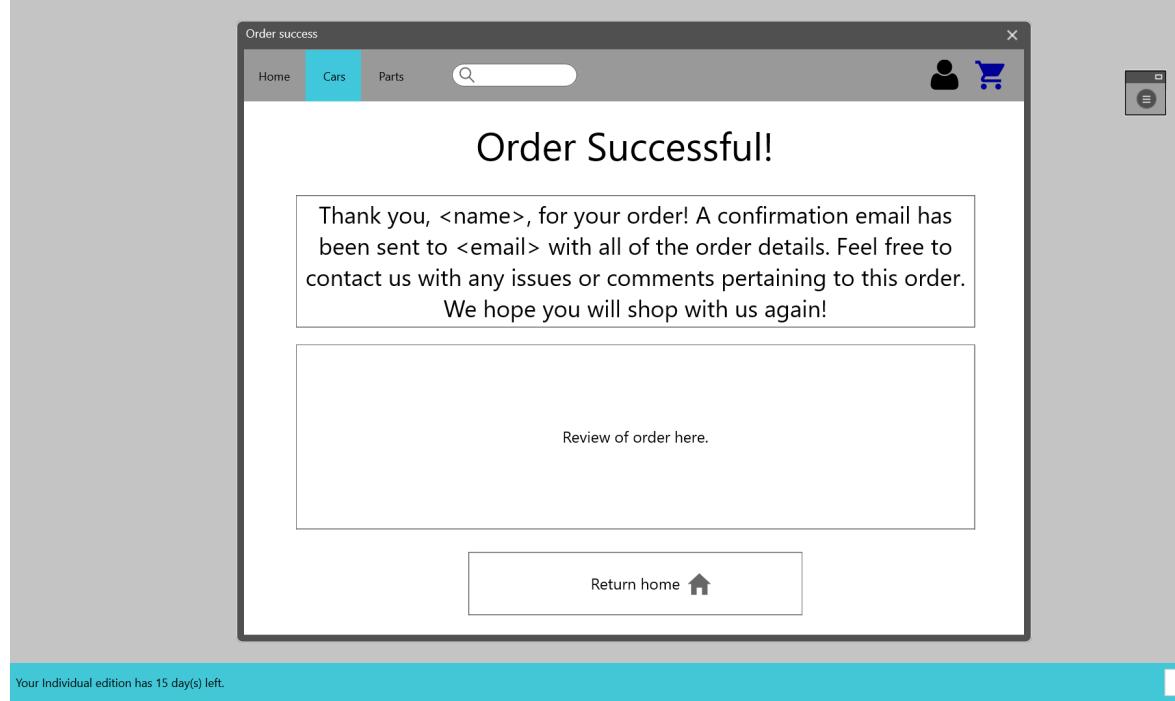
Once a user has one or more products in their cart, they are able to checkout and purchase their products. The first step to purchasing their items is to go to the cart page. They can go to the cart page by clicking the shopping cart button that is located on the top right corner of the navigation bar.



In their cart they can see the items that they are about to purchase, and the subtotal for all of the items. They can remove the items that they do not want using the button on the left side of their item. Once they have everything that they want to purchase in their cart, they can checkout by clicking the button below the subtotal.

The screenshot shows a web browser window with a dark grey header bar. In the header, there are three tabs: 'Home' (grey), 'Cars' (blue, indicating it's the active tab), and 'Parts'. To the right of the tabs is a search icon (magnifying glass) and a user profile icon. On the far right of the header is a small window icon. Below the header, the main content area has a white background. At the top of this area, the word 'Checkout:' is centered in a bold black font. Below 'Checkout:', there is a horizontal list of seven input fields, each preceded by a label: 'First name:' (with an empty input field), 'Last name:' (with an empty input field), 'Email:' (with an empty input field), 'Phone number:' (with an empty input field), 'Shipping address:' (with an empty input field), 'Credit card #' (with an empty input field), and 'Card expiration date:' (with an empty input field). At the bottom of the form is a large rectangular button containing the text 'Confirm Purchase' followed by a blue checkmark icon. Below the main content area, there is a thin teal footer bar. On the left side of the footer, the text 'Your Individual edition has 15 day(s) left.' is displayed. On the right side of the footer, there is a small 'Buy now' button.

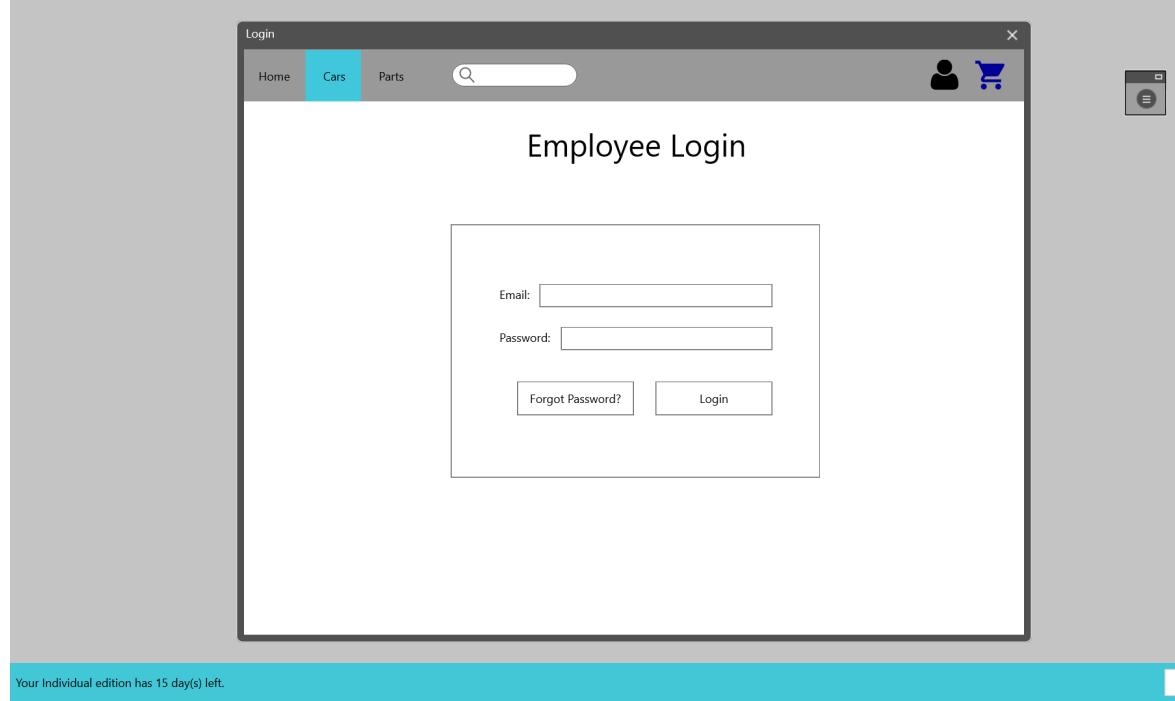
A form will appear that the customer must fill out in order to purchase their products. The information that the customer needs to fill out includes their first name, last name, email, phone number, address, and credit card information. Once they fill out every section, they can click the 'Confirm Purchase' button at the bottom of the page. If the user forgets to fill out a section or if they put incorrect information, then the application will return them to the form and tell them which section is wrong.



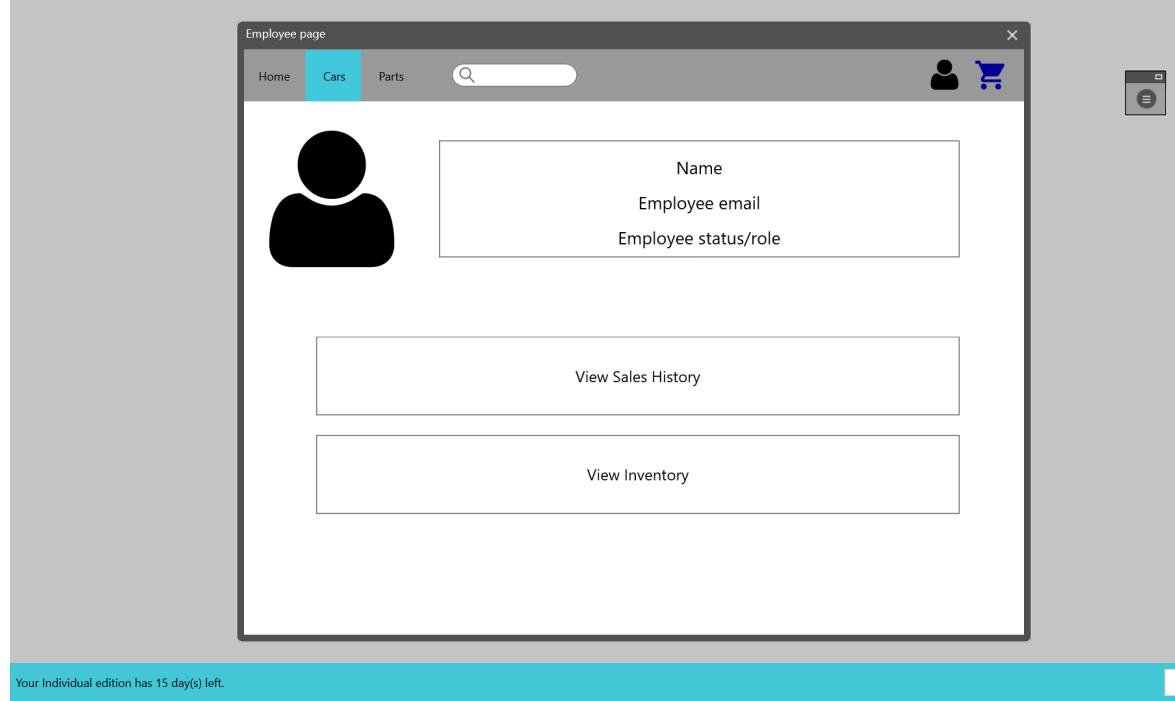
If the customer's order was successfully processed, they will be taken to a page that will thank them for their order. This page also tells them that a confirmation email has been sent to them. Below that will be a review of the customer's order. At the bottom of the page will be a button that a user can click to return to the home page.

### C. UC-10, 11: CrudSalesData, CrudInventoryData

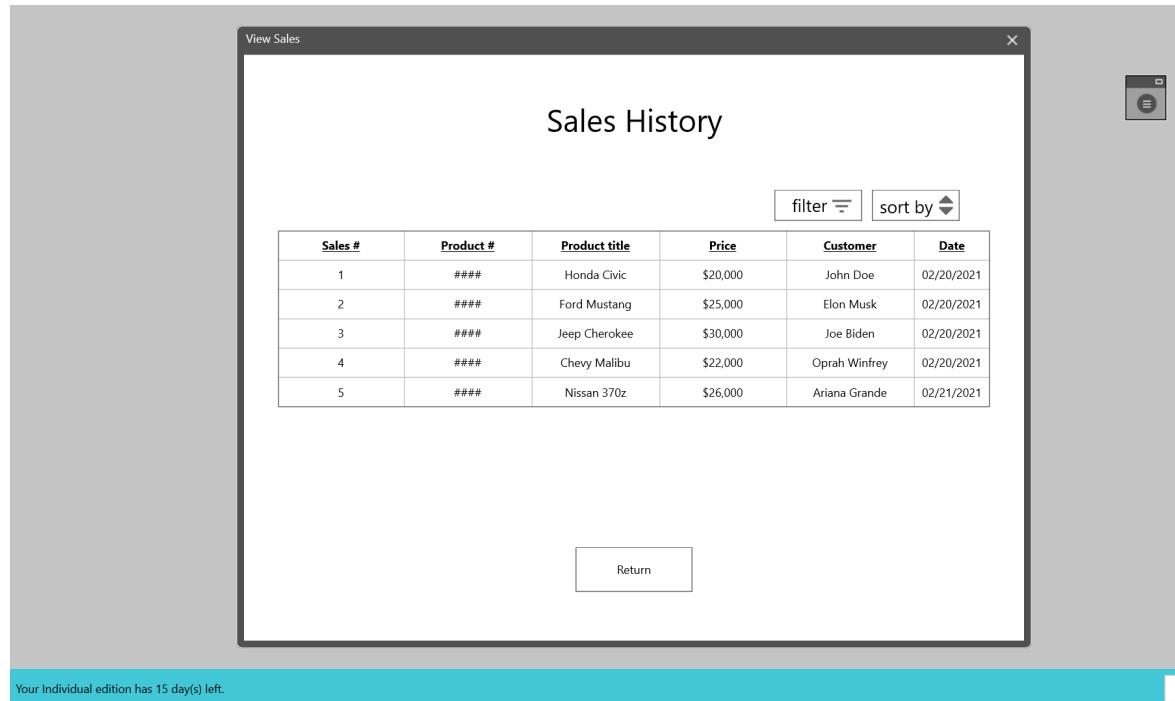
This web application will feature extra features for employees and administration that the customers will not be able to access. If an employee or a member of administration clicks on the login button that is on the navigation bar next to the shopping cart, they will be prompted to login.



The employee will be prompted to enter their email address and the password that are associated with their account. If they do not enter a valid email or password, the web application will tell them that their email or password is incorrect and they can try again. There is a forgot password button where they can have a reset confirmation sent to their email if they click the button. Once they have entered in the correct information, they can hit the login button to sign in. If the web application recognizes their information as an employee account, they will be taken to an employee page.

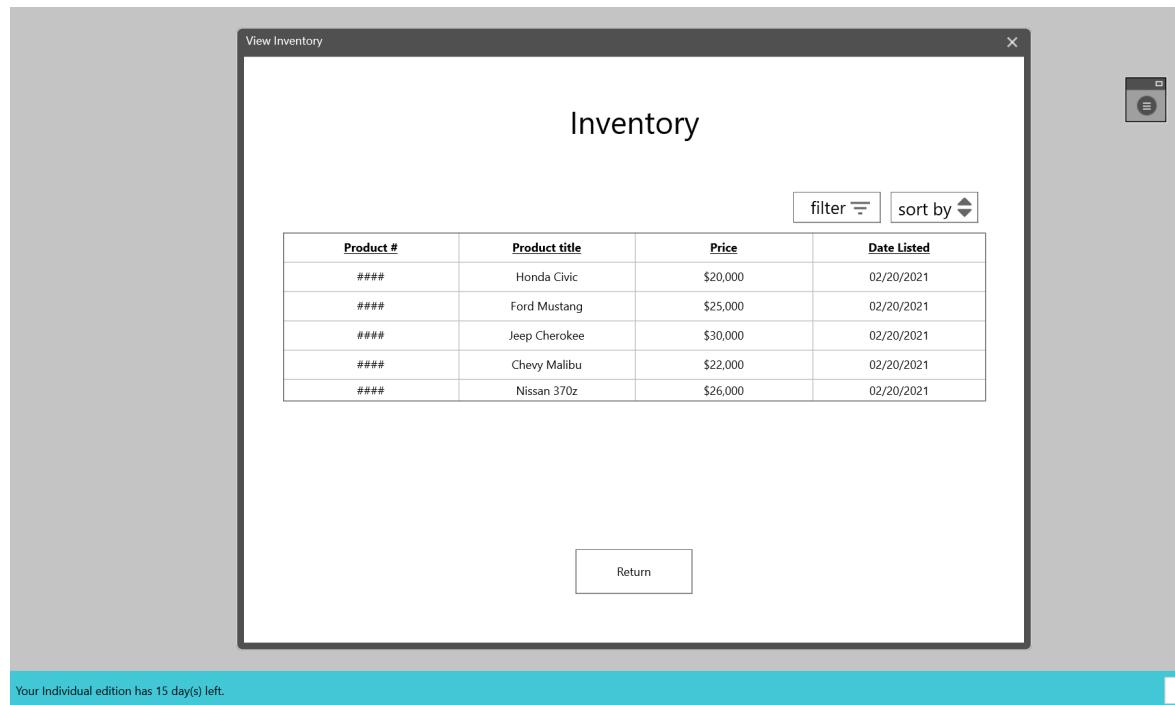


Within this employee page, there are two things that they can do that are not available to regular customers. The first thing that they can do is view sales history. This can be done by clicking on the 'View Sales History' button.



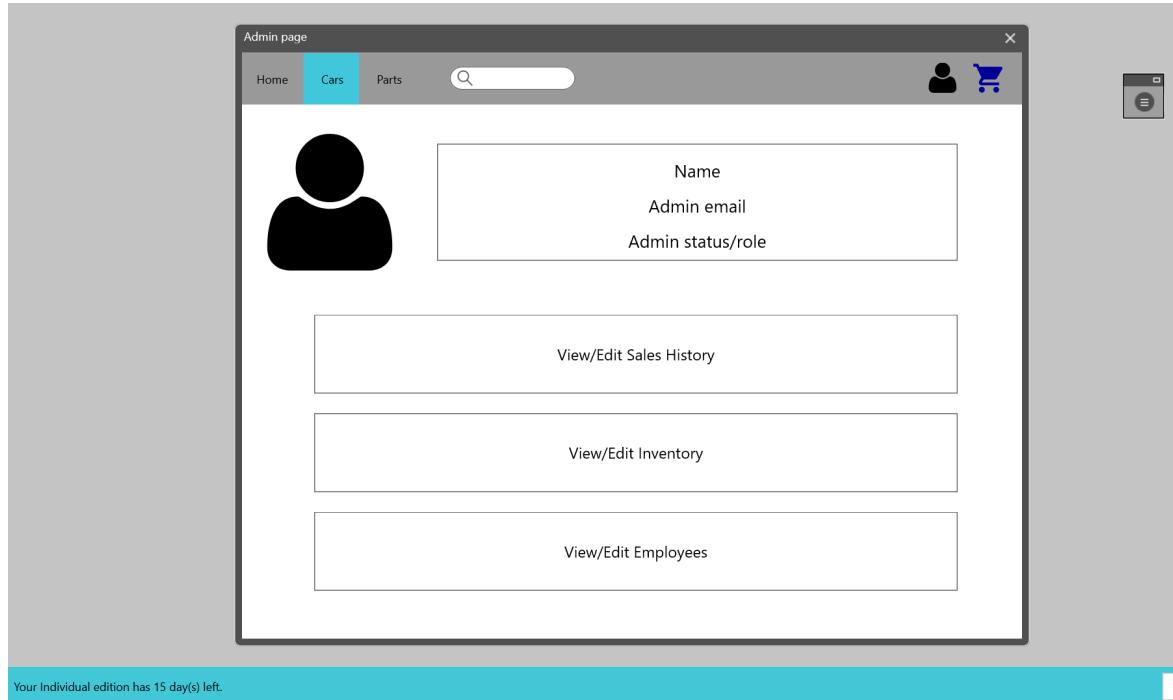
The employee will then see all of the previous sales that have been recorded to the database. This data will be in an easy to understand format. They will also have the ability to filter and sort the data as they please to find the specific sales record that they are looking for. Once they are done, they can click the button at the bottom of the page to return to their employee page.

An employee can also view all of the current inventory in the database by clicking on the 'View Inventory' button.

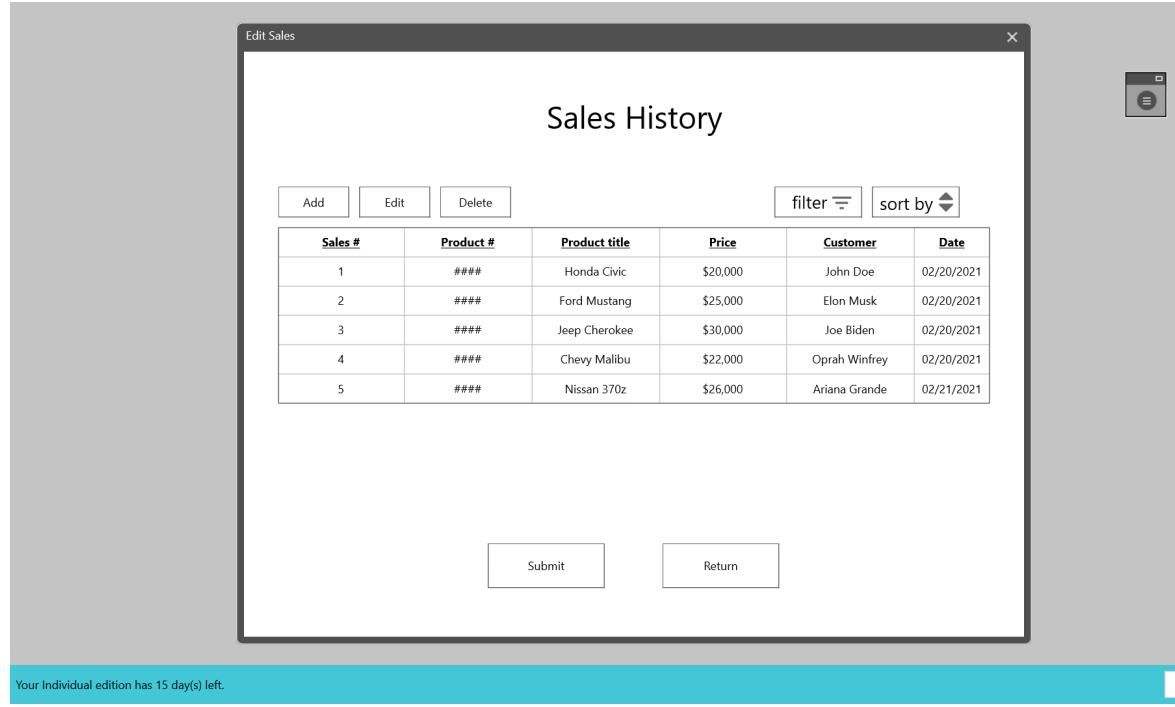


This will look similar to the view sales history, except that it will have accurate and updated inventory data rather than sales history. Once again, it will be in an easy to read format that the user will be able to filter and sort through. Once they are done, they can click the return button at the end of the page.

If the email and password that was provided at the login screen is recognized as being associated with an administrator account, they will be taken to a slightly different administrator page that has added features.



The first thing that an administrator is able to do is view and modify the sales history database by clicking on the top button that says 'View/Edit Sales History'.



Your Individual edition has 15 day(s) left.

Buy now

This will take them to the page that is similar to the sales history page of the employees, but the administrator will have the option to add, remove, or change the contents in it by clicking on the new buttons above the table. They will be able to do these tasks with no knowledge of coding or databases. Once they have modified the data to their liking, they can click the new ‘Submit’ button that is at the bottom of the page. The web application will verify these changes before they are submitted to the database. If the changes are able to get verified, the administrator will receive a message letting them know and the changes are added to the database. If the changes are not able to be verified, the database will not change and the administrator will be informed that their changes could not be submitted.

Back at the administrator page, the second button that they can click on is the ‘View/Edit Inventory’ button. This will take them to a page that allows them to view or modify the product inventory.

The screenshot shows a web application window titled 'Edit Inventory'. The main title of the page is 'Inventory'. At the top, there are four buttons: 'Add', 'Edit', 'Delete', and a group of 'filter' and 'sort by' buttons. Below these is a table with four columns: 'Product #', 'Product title', 'Price', and 'Date Listed'. The table contains six rows of data. At the bottom of the page are two buttons: 'Submit' and 'Return'. A status bar at the bottom left says 'Your Individual edition has 15 day(s) left.' and a 'Buy now' button is on the right.

Product #	Product title	Price	Date Listed
####	Honda Civic	\$20,000	02/20/2021
####	Ford Mustang	\$25,000	02/20/2021
####	Jeep Cherokee	\$30,000	02/20/2021
####	Chevy Malibu	\$22,000	02/20/2021
####	Nissan 370z	\$26,000	02/20/2021

This is very similar to the view/edit sales history page except that it lists the products in the database instead. Once again, the administrator can add, change, or delete entries in this table by clicking on the corresponding buttons above the table. They can click the submit button at the bottom of the page and the web application will verify these changes. If they are approved, the changes are made to the database and the administrator is notified. If they are not approved, the database does not change and the administrator is notified that their changes could not be approved.

The third thing that an administrator is able to do on their page is view and edit the current employees at the dealership. This can be done by clicking on the last button on their administration page, ‘View/Edit Employees’.

The screenshot shows a software window titled "Edit Employee" with a header bar and a toolbar below it. The main area is titled "Employees" and contains a table with six rows of employee data. The table has four columns: "Employee #", "Name", "Job Title", and "Date Hired". The data is as follows:

Employee #	Name	Job Title	Date Hired
####	John Doe	Sales Manager	02/20/2021
####	Jane Doe	Sales	02/20/2021
####	Leeroy Jenkins	Customer Service	02/20/2021
####	Morgan Freeman	Marketing Manager	02/20/2021
####	Adam Sandler	Data Analyst	02/20/2021

At the bottom of the window, there are two buttons: "Submit" and "Return". A status bar at the bottom left says "Your Individual edition has 15 day(s) left." and a "Buy now" button is on the right.

They will then be taken to a page that lists all of the employees that are in our database. The administrator can view, add, edit, and delete employee information. They are also able to sort and filter the employee records to find a specific employee easier. Once they have modified the records, they can click the submit button at the bottom of the page to have their changes verified and receive a message about if the changes were approved or not.

## **II. User effort estimation**

### A. View car information

1. Navigation: total 2 clicks required, 4 optional clicks

- a) Click ‘Cars’ tab
- b) Click desired car card
- c) Optional: Click ‘Filter’ tab
- d) Optional: Click filter requirement
- e) Optional: Click ‘Sort’ tab
- f) Optional: click sort requirements

2. Data Entry: No data entry required

B. View sales history/inventory

- 1. Navigation: total 4 clicks required
  - a) Click log in tab at the top of the page
  - b) Click on email field
  - c) Click ‘Login’ button
  - d) Click ‘View Sales History’ or ‘View Inventory’ button
- 2. Data Entry: total 36 keystrokes required
  - a) Enter email address (average 25 keystrokes)
  - b) Press tab to move to password field
  - c) Enter password (average 10 keystrokes)

C. Purchase an item

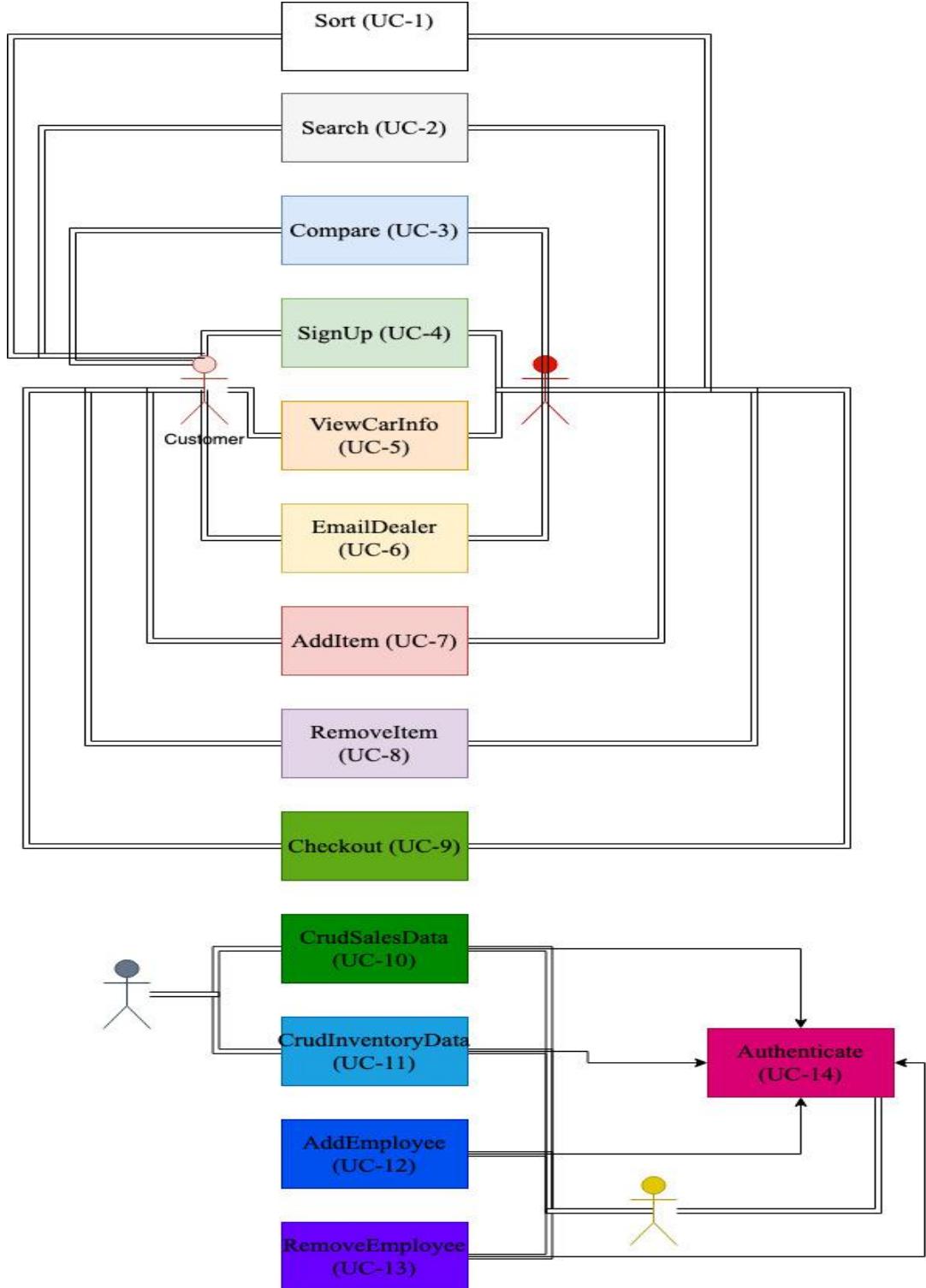
- 1. Navigation: total 6 clicks required
  - a) Click ‘Cars’ or ‘Parts’ tab on navigation bar
  - b) Click the plus button to add to cart
  - c) Click the shopping cart button on the navigation bar
  - d) Click the checkout button
  - e) Click the ‘First name’ Field
  - f) Click the confirm purchase button
- 2. Data Entry: total 110 keystrokes required
  - a) Enter first name (average 6 keystrokes)
  - b) Press tab to move to last name
  - c) Enter last name (average 6 keystrokes)
  - d) Press tab to move to email
  - e) Enter email (average 25 keystrokes)
  - f) Press tab to move to phone number
  - g) Enter phone number (10 keystrokes)

- h) Press tab to move to shipping address
- i) Enter shipping address (average 36 keystrokes)
- j) Press tab to move to credit card #
- k) Enter credit card #(16 keystrokes)
- l) Press tab to move to card expiration date
- m) Enter card expiration date (5 keystrokes)

D. Search for specific product

- 1. Navigation: total 1 click required
  - a) Click the search bar in the navigation menu
- 2. Data entry: maximum of 41 keystrokes required
  - a) Type the desired product name (40 keystrokes max)
  - b) Hit enter to search

# System Architecture



## 1. Architecture Styles

The architectural style we are using I would describe as **The important stuff—whatever that is:** it's concerned with the systems and scenarios that are of paramount importance to the application.

## 2. Global Control Flow

This will be an event-driven application where a loop will be triggered by an action and generated in a different order.

## 3. Time dependency:

I would describe it as real-time handling the customer database and purchases. Updating car models and figures, prices, etc., can be done periodically.

# Project size (Use Case Points)

## UAW

Actor	Description	Complexity (1-3)
Customer	Customers are interacting with the application through an online brower.	3
Employee	Employees are interacting with the application through an online brower.	3
Admin	Admin are interacting with the application through an online brower.	3
Database	Database is interacting with our application through sending/receiving data.	2

**Total UAW = 11**

## UUCP

Use Case	Description	Complexity	Weight
Sort (UC-1)	Complex user interface. Four steps for the main success scenario. Two participating actors (Application, Database).	Average	10
Search (UC-2)	Complex user interface. Four steps for the main success scenario. Two participating actors (Application, Database).	Average	10

Compare (UC-3)	Complex user interface. Five steps for the main success scenario. Two participating actors (Application, Database).	Complex	15
SignUp (UC-4)	Moderate user interface. Six steps for the main success scenario. Two participating actors (Application, Database).	Average	10
ViewCarInfo (UC-5)	Simple user interface. Four steps for the main success scenario. Two participating actors (Application, Database).	Simple	5
EmailDealer (UC-6)	Moderate user interface. Four steps for the main success scenario. One participating actor (Application).	Simple	5
AddItem (UC-7)	Complex user interface. Six steps for the main success scenario. Two participating actors (Application, Database).	Complex	15
RemoveItem (UC-8)	Complex user interface. Six steps for the main success scenario. Two participating actors (Application, Database).	Complex	15
Checkout (UC-9)	Complex user interface. Seven steps for the main success scenario. Two participating actors (Application, Database).	Complex	15
CrudSalesData (UC-10)	Complex user interface. Four steps for the main success scenario. Two participating actors (Application, Database).	Complex	15

CrudInventoryData (UC-11)	Complex user interface. Four steps for the main success scenario. Two participating actors (Application, Database).	Complex	15
AddEmployee (UC-12)	Moderate user interface. Four steps for the main success scenario. Two participating actors (Application, Database).	Average	10
RemoveEmployee (UC-13)	Moderate user interface. Four steps for the main success scenario. Two participating actors (Application, Database).	Average	10
Authenticate (UC-14)	Simple user interface. Five steps for the main success scenario. Two participating actors (Application, Database).	Simple	5

**Total UUCP = 155**

### TCP

Technical Factor	Description	Weight	Perceived Complexity	Calculation (W * PC)
T1	Distributed, Web application	2	3	6
T2	Users expect quick, but speed is not everything	1	2	1
T3	Users expect perfect efficiency	1	5	5
T4	Internal application workings are complicated	1	4	4
T5	Average reusability	1	3	3
T6	No install	0.5	0	0

T7	Ease of use is of primary importance	0.5	5	0
T8	Ability to keep database options open	2	1	2
T9	Additional features and updates are likely	1	4	4
T10	Concurrent use is required	1	1	1
T11	Security is challenging to implement, and very important	1	5	5
T12	No third party access	1	0	0
T13	No training needed	1	0	0

**Technical Factor Total = 31**

$$\text{TCF} = 0.6 + (0.01 * 31) = 0.91$$

$$\text{UCP} = (11 + 155) * 0.91 = 151.06$$

# Project Management

## Upcoming Due Dates:

Interaction Diagrams: 03/01/2021

Class Diagram and System Architecture: 03/08/2021

Report 2: 03/15/2021

## Plan of Work:

Finish Individual Work on Interaction Diagrams: 02/26/2021

Group Meeting: 03/01/2021 (Divide up work for Class Diagram and System Architecture)

Finish Individual Work on Class Diagram and System Architecture: 03/04/2021

Group Meeting: 03/08/2021 (Divide up final work for Report 2)

Finish Individual Work on Report 2: 03/11/2021

Full effort into finishing all high level requirements for project (leave a few for demo 2)

First Demo Presentation: 04/05/2021

Report 3 - Part 1: 04/12/2021

Report 3 - Part 2: 04/19/2021

Second Demo Presentation: 05/03/2021

## References

V12data.com. *90% of Car Shoppers Want an Buying Process Online | V12*. [online] Available at: <https://v12data.com/blog/90-car-shoppers-prefer-dealership-where-they-can-start-buying-process-online/>. [Accessed 8 Feb. 2021].

Tutorialspoint.com. (2019). *Software Requirements*. [online] Available at: [https://www.tutorialspoint.com/software\\_engineering/software\\_requirements.htm#:~:text=There](https://www.tutorialspoint.com/software_engineering/software_requirements.htm#:~:text=There) [Accessed 10 Feb. 2021].

www.tutorialride.com. (n.d.). *User Interface*. [online] Available at: <https://www.tutorialride.com/software-architecture-and-design/user-interface.htm#:~:text=There%20are%20eight%20characteristics%20considered%20while%20making%20a> [Accessed 10 Feb. 2021].

www.bestbuy.com. (n.d.). *Best Buy | Official Online Store | Shop Now & Save*. [online] Available at: <https://www.bestbuy.com>. [Accessed 10 Feb. 2021].

www.bigoautollc.com. (n.d.). *Big O Auto* [online] Available at: <https://www.bigoautollc.com/details/used-2016-acura-tlx/71123677>. [Accessed 10 Feb. 2021].

docs.erpnext.com. (n.d.). *Product Page*. [online] Available at:  
<https://docs.erpnext.com/docs/user/manual/en/website/product-page> [Accessed 10 Feb. 2021].

ReviewPro. (n.d.). *mock-up-desktop-and-mobile*. [online] Available at:  
<https://www.reviewpro.com/blog/reviewpro-continues-innovate-guest-experience-improvement/mock-up-desktop-and-mobile/> [Accessed 10 Feb. 2021].

Saring, J. (2019). *5 Ways to Improve UI Consistency*. [online] Medium. Available at:  
<https://blog.bitsrc.io/5-ways-to-improve-ui-consistency-99013bf20417> [Accessed 10 Feb. 2021].

Apple. (2011). *iPhone*. [online] Available at: <https://www.apple.com/iphone/>. [Accessed 10 Feb. 2021].

SolarWinds. (2015). *Web Application Process Flow in Oracle Databases*. [online] Available at:  
<https://logicalread.com/application-flow-oracle-mc08/#.YCfXl2hKjb0> [Accessed 10 Feb. 2021].

cas.fhsu.edu. (n.d.). *Login - CAS – Central Authentication Service*. [online] Available at:  
<https://blackboard.fhsu.edu/> [Accessed 10 Feb. 2021].

Note: A glossary was excluded as it was deemed unnecessary for this submission by the group.  
The language used was lacking technical jargon and anything that might be considered complex.  
A glossary will be added in future submissions as more complicated terms are introduced.