WEBD 335 Final Exam

Project Plan

Michelle Ackley

WEBD 335 Advanced Client-Side Applications

Mr. Clark Powell

Dr. Bradley Watson

December 15, 2018

WEBD 335 Final Exam

Project Plan

# Project Justification

The primary justification / objective of this project is to create and deploy an automotive retail web application at a development and operational cost sufficiently low to guarantee a return on investment of $500,000,000 in the first year and in excess of $1,000,000,000 in year subsequent year of a projected 10-year operational life span. The application will be used by retail buyers and leasers of automobiles of all types, both new and used, in lieu of visiting automobile dealers’ lots in person. Specifically, the proposed application will allow users to replicate in detail the sensory experiences of a buyer when inspecting a particular candidate vehicle, including getting into it, driving it, riding as a passenger, and determining the condition of the interior, exterior, electronic and mechanical systems. Such buyers will be able to do this across a large selection of dealerships, brands, and types of vehicles in all parts of the United States, without ever leaving their place of choice for using the applications.

**Project Description**

The project will be done by two separate project teams, each under separate management, coordinated by the Chief Project Manager (CPM). Team Server-Side will develop those aspects of the application that will reside on the servers. Team Client-Side will develop the user interface aspects of the application that will reside on the end users’ systems of choice. Through a variety of communication channels chosen, operated and controlled by CPM and coordination staff reporting to the CPM, the two project teams will be in close communication, sharing all necessary aspects of the two parts of the project, insuring that the server-side and the client-side parts of the target automotive retail application work flawlessly together in all respects.

**Methodology**

## Team Server-Side Contribution

**Assumptions & Requirements**:

* Server-side already has all required hardware, software, and network capabilities needed for the web application environment
* Client-side framework: AngularJS
  + fast app builder, minimal code required, free, scalability, cleaner
* Server-side API & framework: PHP
  + Well known, open source, works well with AngularJS,
  + PHP will store, get, and modify the data in the server
* The client-side framework (AngularJS) is separate from the server-side API (PHP), meaning that either could be changed at any given time without the other being changed.
* Xampp along with apache will create the web server
* MySQL will be used as the database
* There will need to be 4 tables created in MySQL, Cars, Dealerships, Users, and Favorites
* The server-side will be responsible for its own validation for security and safety, since client-side alone cannot secure user input
* Header(“Access-Control-Allow-Origin: \*”); will be add to PHP code to allow cross-site access

## Team Client-Side Contribution

* Will include all of the views that the end-user will see
* Have basic client-side validation from AngularJS
* Along with the security and validation from the server-side, AngularJS updates and notifies the end-user about the current state of a form or input field. It can tell them if it has been touched, modified, or not touched at all.

Ex. If they didn’t complete a text field that is required, it will notify the user to complete that field, this can be used to help validate that when the dealer is logging in the put in the required fields to login.

<input name="myUsername" ng-model="myUsername" required>

<span ng-show="myForm.myUsername.$touched && myForm.myUser.$invalid">The username is required.</span>

* AngularJS will use this code to be able to access PHP and MySQL:

<script>  
var app = angular.module('carApp', []);  
app.controller('customersCtrl', function($scope, $http) {  
    $http.get("customers\_mysql.php")  
    .then(function (response) {$scope.names = response.data.records;});  
});  
</script>

**Client-Side Requirements Overview**

* **Branding Image for Application**-Car graphic
* **URL:** www.carsnow.com
* **Index-**View homepage place to select car searching criteria or sign in if you are a dealer
* **Cars-**View list of cars that fit search criteria, with option to filter and sort list
* **Favs-**View list of cars selected from cars.html
* **Dealers-**View dealers can access to view/adjust their car lot’s inventory
* **Functionality for all types of users**-Car Buyers, Car Dealers,

**Client-Side Requirement in Detail**

**Branding Image:**

* Square-Solid Black
* Image of Car-White, outlined in red; Centered on square
* Text-Cars(white), Now(red) Italic, Arial Font

**Index View-Homepage Detailed Design:**

1. **Header:**
   1. Cars Now logo-centered
   2. Dealer Sign in-aligned right, with down caret
   3. Sign in caret- will dropdown a sign in form
   4. Form-Two user input boxes for username and pw
   5. Form-Submit button called Sign-In, if clicked will go to dealer’s view
2. **Body:** 
   1. Form-centered
   2. Form Title-Ready to find your next car?
   3. Form-5 areas for user input, make, model, max-price, max-distance, new/used
   4. Form-make, dropdown list of all makes
   5. Form-model, dropdown list of all models of the make selected above
   6. Form-max price, dropdown list of increments of $5,000
   7. Form-max distance, dropdown list with increments of 25 miles
   8. Form-new/used toggle
   9. Form-submit button, Search takes user to car’s view

**Cars View Detailed Design:**

1. **Header:**
   1. Cars Now Logo-centered, if clicked will return to Index/Home
   2. Sort/Filter-aligned right, Sort | Filter, if clicked sort/filter will prompt a popover list
   3. Sort dropdown- Car Price: High v, Low ^, Car Year: New->Old, Old->New
   4. Filter popover-headings with dropdown carets: year, make, model, price, mileage, color, drivetrain
   5. Filter caret dropdown-if a heading is clicked it will prompt a dropdown with check boxes for users to make specific selections like Color v: Black, Red, Blue, Gray, White, Yellow
2. **Body:**
   1. Table-List of 50 cars that match user criteria
   2. Table Element-Car exterior image, year, make, model, price, mileage, carat down icon, star icon, an X
   3. Star icon-will be gray, when clicked it will turn yellow and car will be added to favs view, color needs to toggle back and forth on clicks
   4. The X-will be gray, when hovered over it will turn red, when clicked it will remove car from list
   5. Car image-when clicked it will open a pop-up box with more images of car, and a 3D tour of the car
   6. Caret down- when clicked will expand the car table element and list more details of the car like dealer, drivetrain, clean CarFax y/n?, additional features
3. **Footer:**
   1. Star Icon(2)-centered, when clicked will take user to favs view

**Favs View Detailed Design:**

1. **Header:**
   1. Cars Now Logo-centered, if clicked will return to Index/Home
2. **Body:**
   1. Table-List of cars that user selected as their favorite
   2. Table Element-Car exterior image, year, make, model, price, mileage, carat down icon, star icon, Contact Dealer Now button
   3. Car image-when clicked it will open a pop-up box with more images of car, and a 3D tour of the car
   4. Caret down- when clicked will expand the car table element and list more details of the car like dealer, drivetrain, clean CarFax y/n?, additional features
   5. Star icon-will be highlighted yellow, but if clicked will be removed from favs list
   6. Contact Dealer Button-when clicked will pop up the dealerships phone number to contact them for availability
3. **Footer:**
   1. Car graphic-centered, when clicked will take users back to cars view

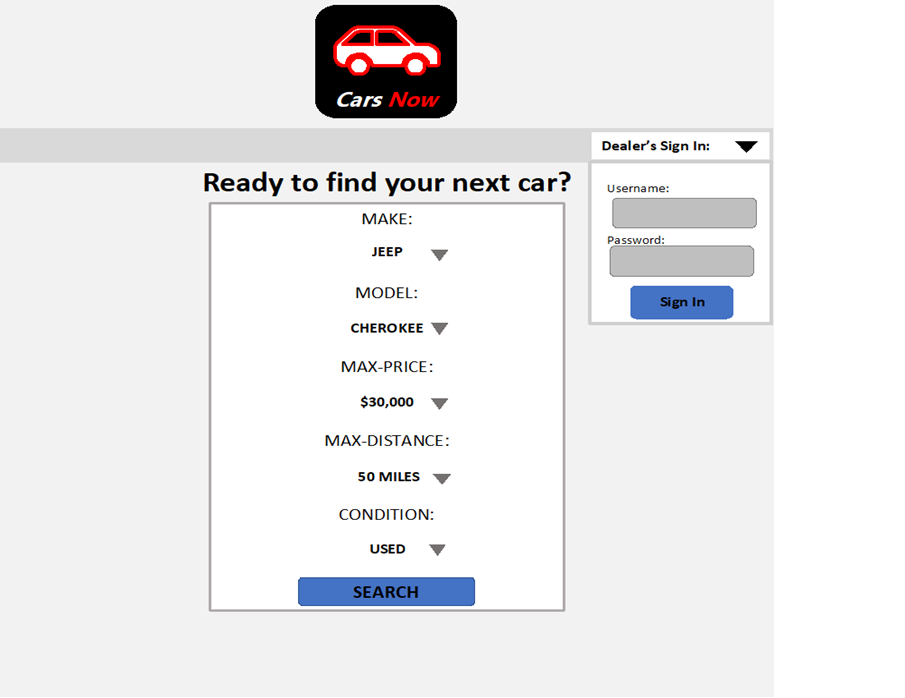
**Dealer’s View Detailed Design:**

1. **Header:**
   1. Cars Now Logo-centered, if clicked will return to Index/Home
   2. Add icon-aligned right, if clicked will dropdown a form for adding a new car
   3. Car form-A form will ask for year, make, model, price, mileage, an image, and any additional features
2. **Body:**
   1. Table-List of all dealer’s inventory
   2. Table element-will show all details and features of each car, with edit and X icons
   3. Edit icon-when clicked will show text area for each category of the car and an add button to add additional features, and a cancel, and save button
   4. X icon-will remove that car from the dealership’s inventory
3. **Footer:**
   1. 2 Objects- Cars sold, Cars on Lot
   2. Cars sold-will have the number of cars that were sold/removed from inventory list
   3. Cars on lot-will have the current number of cars that are the lot available to buy

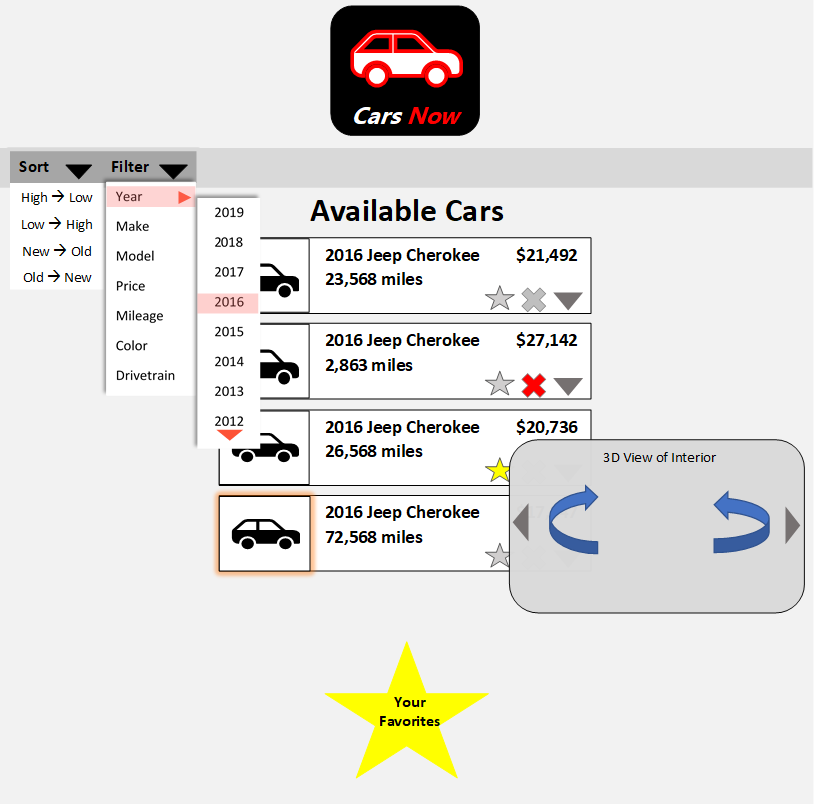
**Agile/Scrum Methodology**

* Agile is the best approach to make the automotive retail application
* Agile is very business side, and allows you to address a client’s needs right away instead of waiting for the next step in the process
* **To make a ROI of $500,000,000 there is no time to wait for process, the client’s company needs deliverables immediately.**
* Since there are two distinguished small teams it is critical that they are cross-functional. That means that there needs to be consistent communication and coordination between the two from the beginning.
* The Scrum Master/CPM is the head communicator among the entire group, to ensure seamless communication is happening within the group.
* Agile will keep team members accountable because of the daily Scrum meetings set by the Scrum Master.
* To help recognize the exact requirements needed we implemented a fictional client in the design images, to ensure that a product requirement is not missed.
* In part one, I created a product log that lists each item we need to create for each page and what is required for the application to be finished. The log makes for a great visual to show team progression especially for “sprints”.
* The projects requirements are divided up into sprints and given a set deadline the sprint needs to be accomplished. Agile performs “sprints” throughout the entire project.
* After each sprint our team needs to have a shippable increment, that shows a fully-functioning feature, or completed application, this will help streamline productivity, improve worker morale, constant feedback, continuous progress, early delivery, updated needs for hardware requirements
* For software and internet requirements each developer will have access to a modern browser, and an integrated development environment with a debugger like NetBeans.
* I will start with what I am familiar with and what I know worked in the past. After running our first sprint we will see how the internet and software requirements need to be improved.

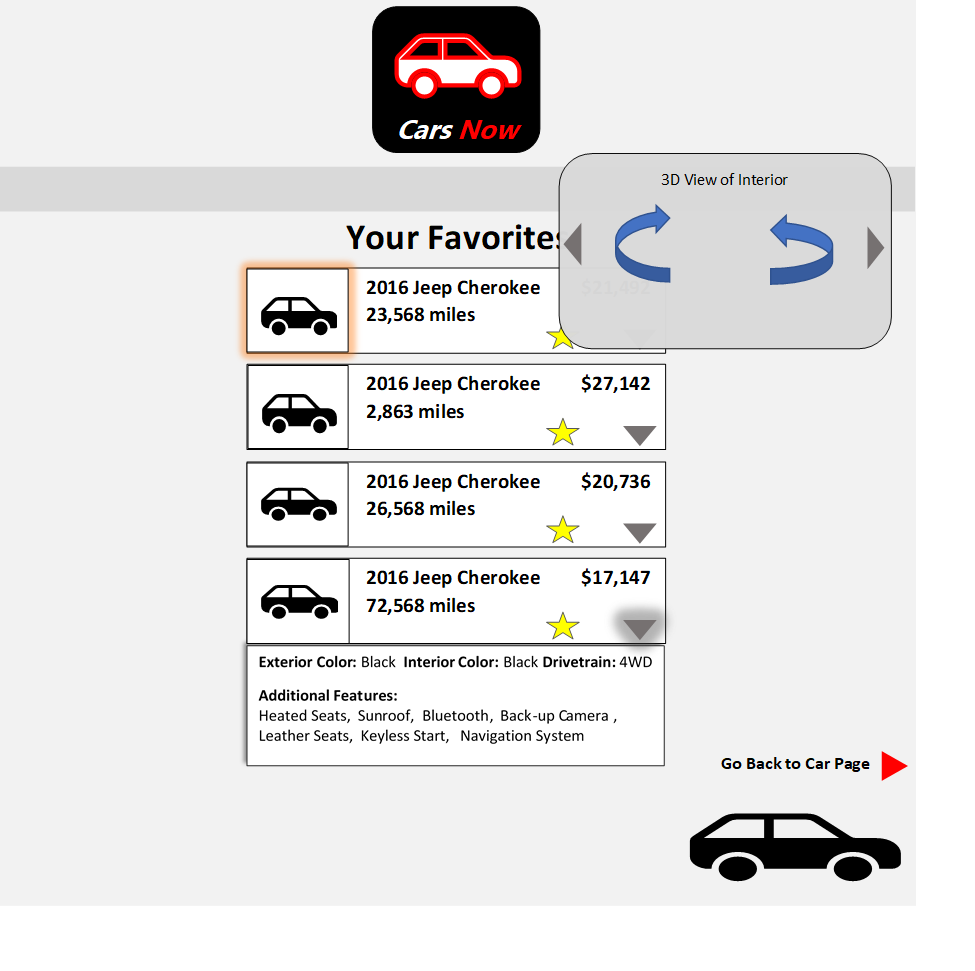
**Index/Homepage View Draft**



**Cars View Draft**



**Favorites View Draft**



**Dealer’s View Draft**

