Design Specification

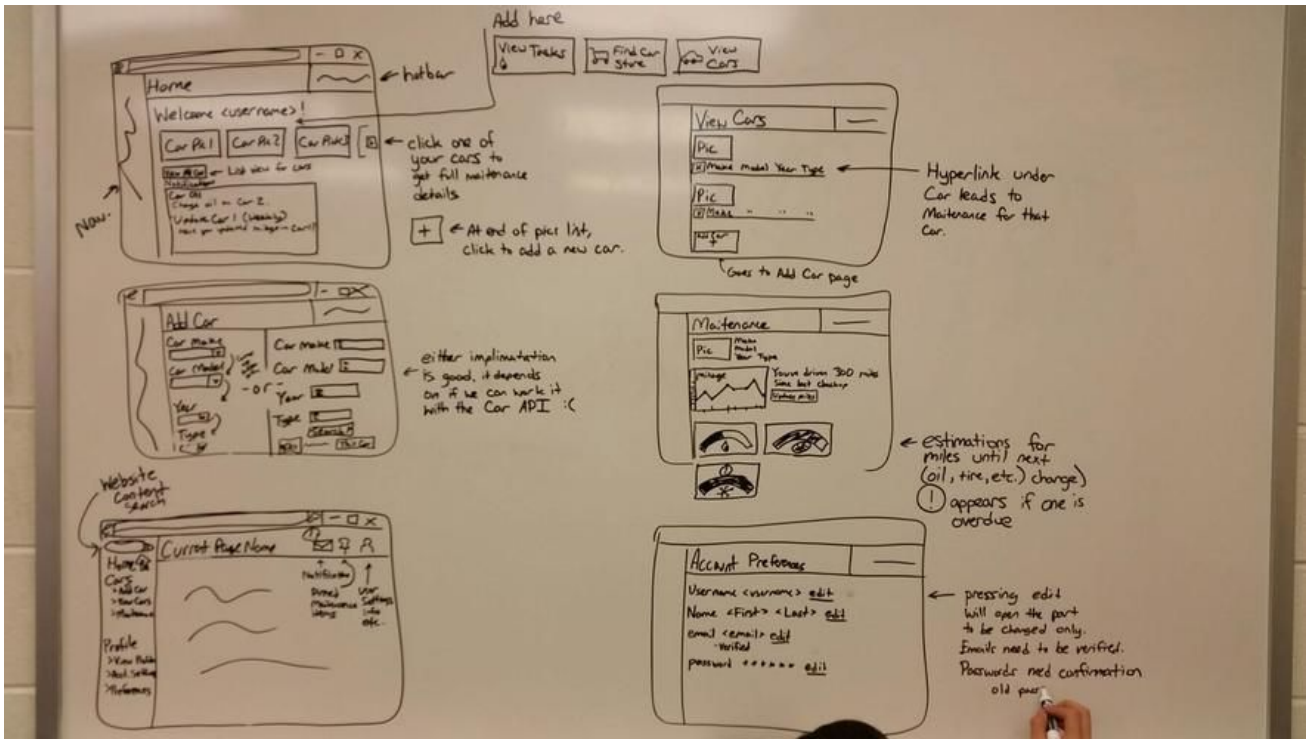
Authors: Michael Moscariello, Sean O'Connell, Joe Mecca, Michael Crinite, Andrew Genova, Matthew Gimbut

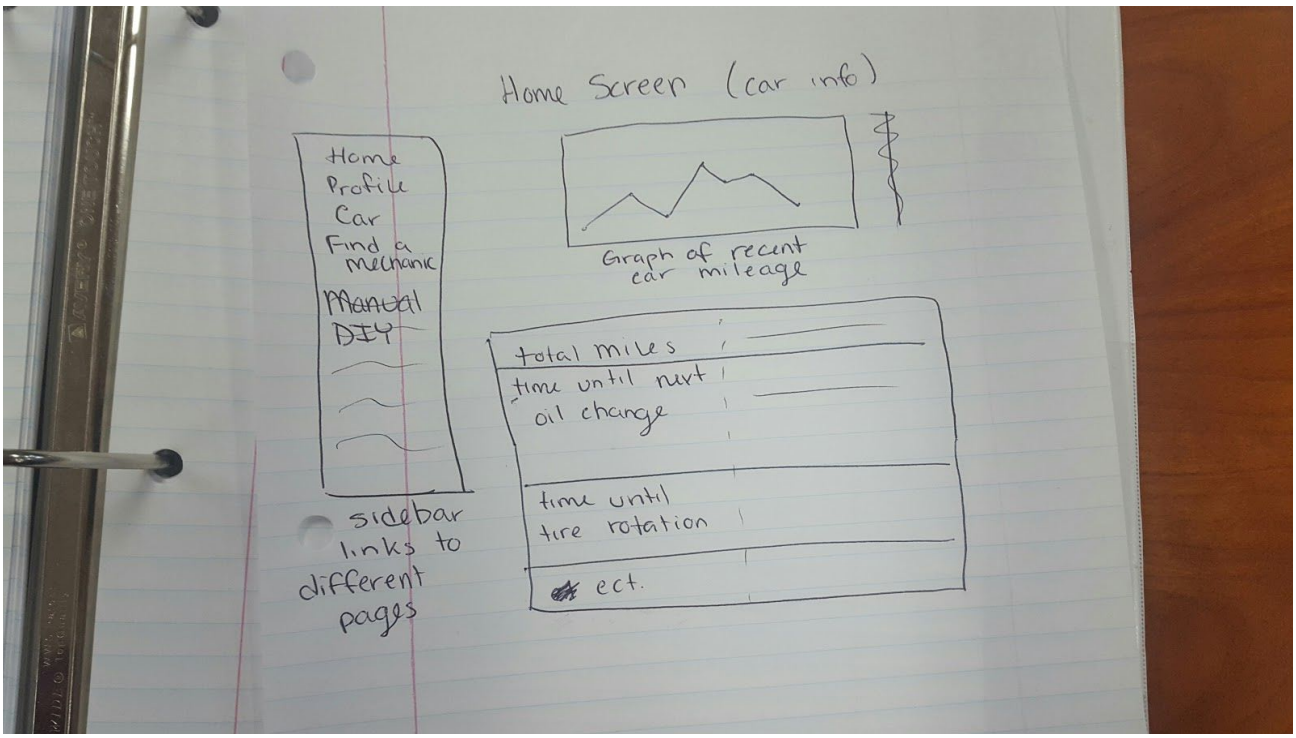
0. SUMMARY

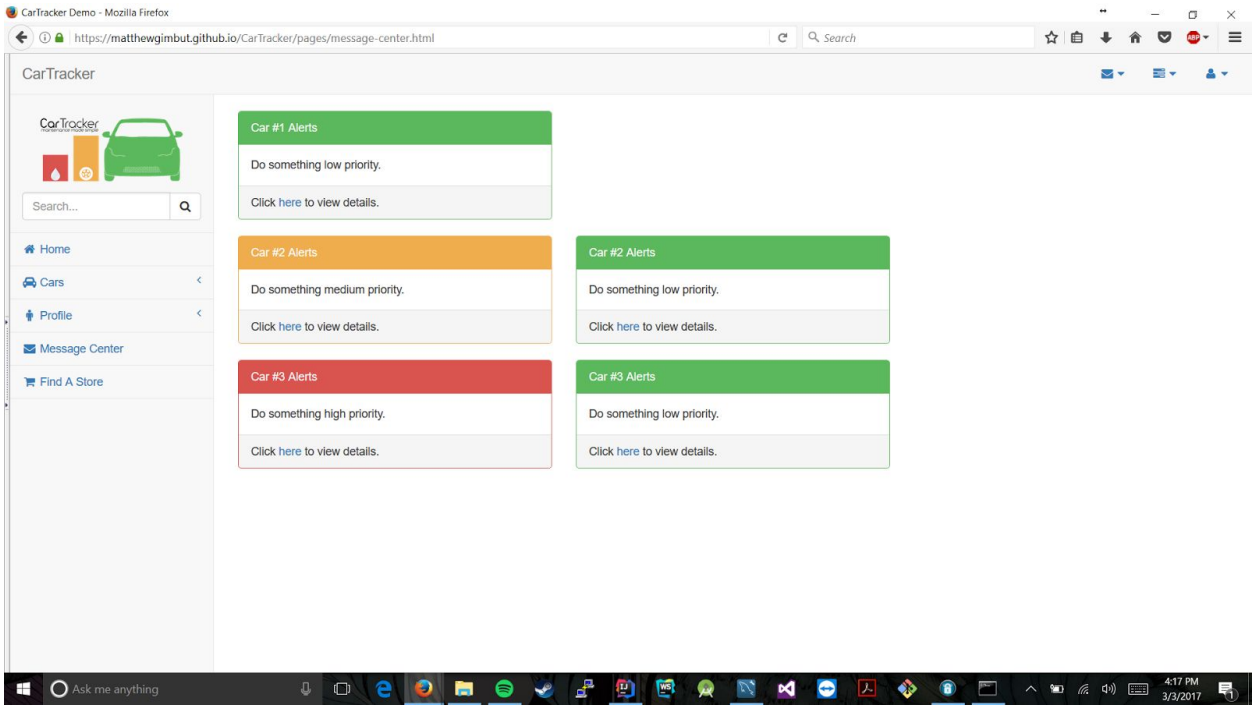
The goal of this product is to provide a sort of guided encyclopedia for car owners who may not be as savvy about the mechanical aspects of their car. The app will take in basic information about the car such as mileage, and recommend maintenance that should be done based on provided information. The app will also include a database of car manuals so that users can learn more about their own cars.

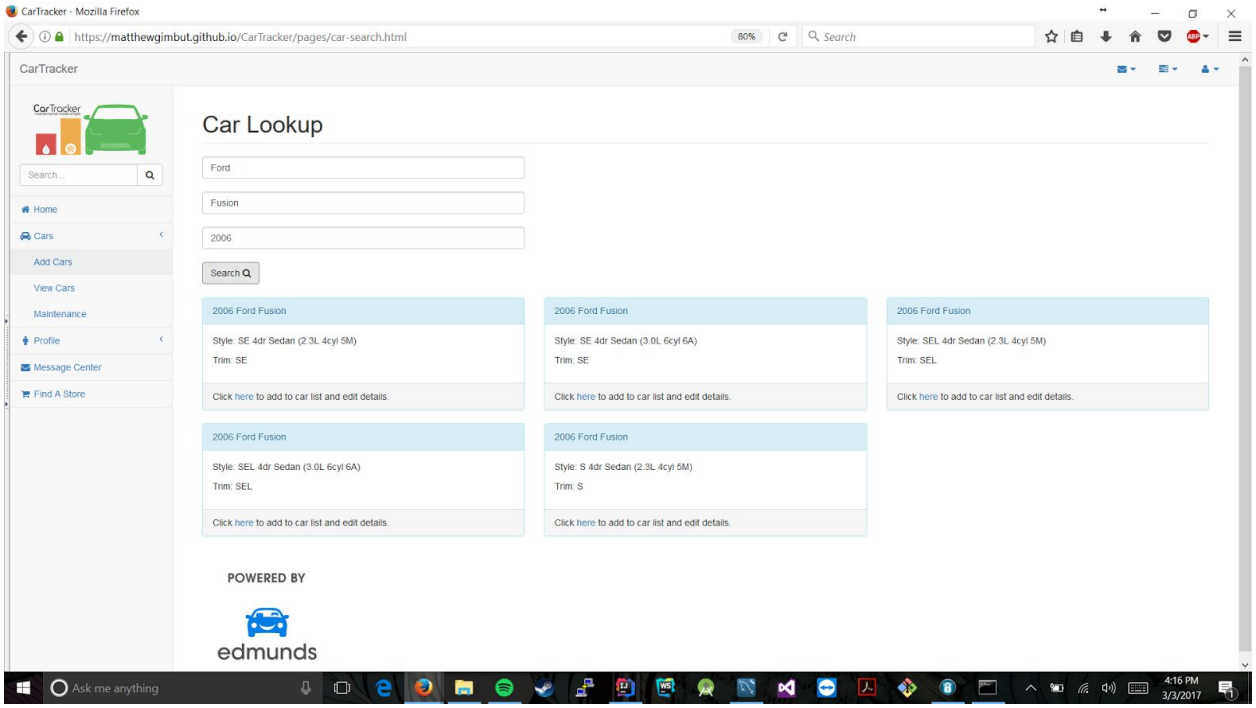
The app will provide the end user with a tool to help them keep organized very easily. Even if they know nothing about cars or how to maintain them, this app will ideally provide a a way to instruct them how in the most convenient way possible.

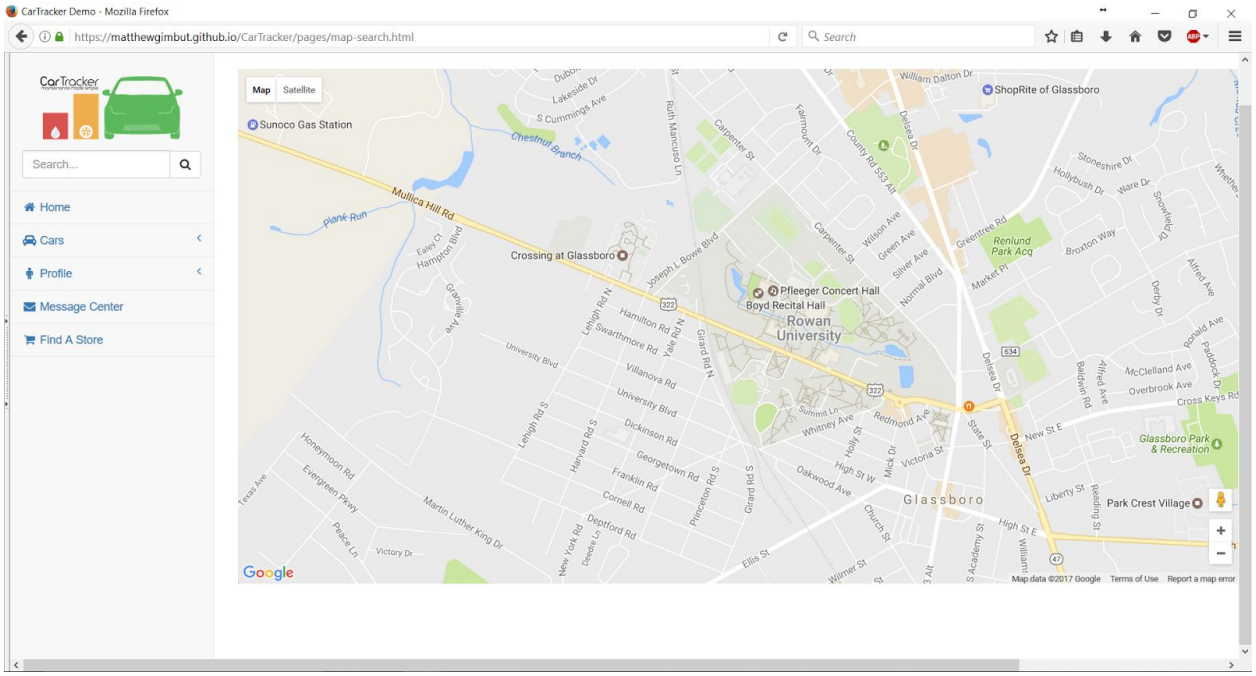
1. SCREEN/VISUAL SECTION:

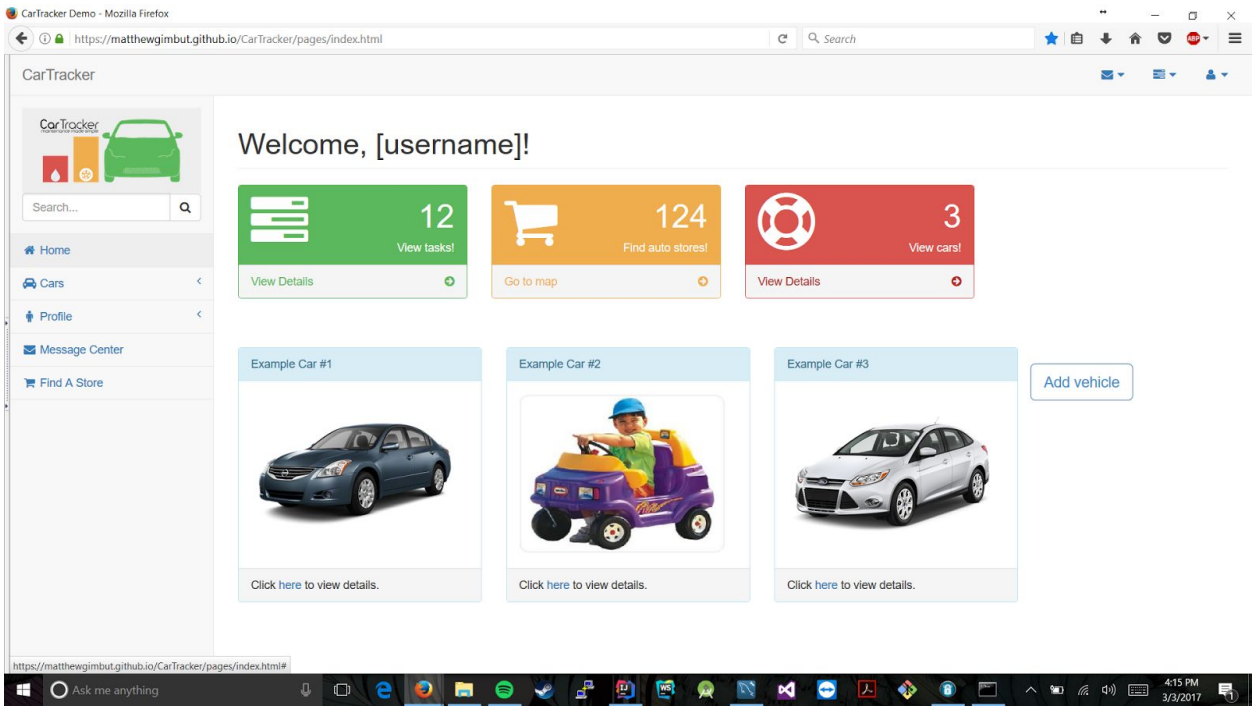



Home Screen (car info)
Home
Profile
Car
Find a mechanic
Manual DIY
sidebar links to different pages
Graph of recent car mileage
total miles
time until next oil change
time until tire rotation
ect.


CarTracker Demo - Mozilla Firefox
https://matthewgimbut.github.io/CarTracker/pages/message-center.html
CarTracker
Search...
Home
Cars
Profile
Message Center
Find A Store
Car #1 Alerts
Do something low priority.
Click here to view details.
Car #2 Alerts
Do something medium priority.
Click here to view details.
Car #2 Alerts
Do something low priority.
Click here to view details.
Car #3 Alerts
Do something high priority.
Click here to view details.
Car #3 Alerts
Do something low priority.
Click here to view details.
Ask me anything
4:17 PM
3/3/2017


CarTracker - Mozilla Firefox
https://matthewgimbut.github.io/CarTracker/pages/car-search.html
CarTracker
Car Lookup
Ford
Fusion
2006
Search
Home
Cars
Add Cars
View Cars
Maintenance
Profile
Message Center
Find A Store
2006 Ford Fusion
Style: SE 4dr Sedan (2.3L 4cyl 5M)
Trim: SE
Click here to add to car list and edit details.
2006 Ford Fusion
Style: SE 4dr Sedan (3.0L 6cyl 6A)
Trim: SE
Click here to add to car list and edit details.
2006 Ford Fusion
Style: SEL 4dr Sedan (2.3L 4cyl 5M)
Trim: SEL
Click here to add to car list and edit details.
2006 Ford Fusion
Style: SEL 4dr Sedan (3.0L 6cyl 6A)
Trim: SEL
Click here to add to car list and edit details.
2006 Ford Fusion
Style: S 4dr Sedan (2.3L 4cyl 5M)
Trim: S
Click here to add to car list and edit details.
POWERED BY
edmunds
Ask me anything
4:16 PM
3/3/2017


CarTracker Demo - Mozilla Firefox
https://matthewgimbut.github.io/CarTracker/pages/map-search.html
Search...
Home
Cars
Profile
Message Center
Find A Store
Map
Satellite
Sunoco Gas Station
ShopRite of Glassboro
Crossing at Glassboro
Pfleeger Concert Hall
Boyd Recital Hall
Rowan University
Glassboro
Park Crest Village
Glassboro Park & Recreation
Mullica Hill Rd
Chestnut Branch
Plank Run
Delsea Dr
Google
Map data ©2017 Google
Terms of Use
Report a map error



2. RESTFUL SERVICE SECTION: All endpoints in the RESTful service

Endpoint	**Purpose**	**Inputs**	**Outputs**	**Implementation**
POST /users	Posts a user object to the database	Path parameter	JSON array of a user's information	Retrieve the information created in our web application and store it 'vehicle' database table
POST /cars/:username	Posts a vehicle for a specific user	Path parameter, username **Body:** user information	JSON array of a vehicle's data from a user	Retrieve the vehicle information from one user using our web application and store it in the 'vehicle' database table
POST /cars/mileage	Posts updated mileage for a user's vehicle	Path parameter **Body:** carID, mileage	JSON array of the mileage of a user's vehicle	Retrieve an updated mileage integer from the web application and store it in the respective field in the 'vehicle'

				database table
GET users/:username	Retrieves the user with the given username	Path parameter, username	JSON array of one user's contents	Retrieve the user from the 'users' database associated with the given username for use in our web application
GET /users	Retrieves all users	Path parameter	JSON array of all users	Retrieve all user information for use in the web application
GET /vehicles/:username	Retrieves all vehicles from the username	Path parameter, username	JSON array of the vehicles owned by a user	Retrieve all vehicles associated with the given username for use in the web application.

3. DATABASE SECTION: All Tables and columns

Tables	**Columns**
Users	int UserID (Primary Key, Non-Null, Unique ID, Auto increment) String FirstName (Non-Null) String LastName String Username (Non-Null) String Password (Hash) (Non-Null) String EmailAddress (Non-Null) int Age String DOB
Vehicles	int UserID(Same as Users) (Primary Key, Non-Null) String Make (Non-Null) String Model (Non-Null) int Year (Non-Null) String style float Mileage (Non-Null) Float mileageLastInspection Date mileage String pictureLocalLink String manualLocalLink

4. IMPLEMENTATION PLAN SECTION: Write an implementation plan for the first two weeks in the implementation phase. This must include all the action items for this time period and an assignment of responsibilities for individual team members. Note that I will use your writeup to assess how you have progress at the midway point in the implementation phase.

Week 1 (3/5 - 3/11):

- ➢ Finish designing the website (Everyone)
- ➢ Allow user to create account and link to database (Michael Moscariello)
- ➢ Error checking for user registration/info (Michael Moscariello)
- ➢ Create a page that asks for information for each car (Sean O'Connell)
- ➢ Begin finding a source for user service manuals for most common vehicles (Michael Crinite)
- ➢ Use the google maps API to pinpoint nearby autostores (Matt Gimbut)
- ➢ Allow user to select and save cars they have searched for (Matt Gimbut)
- ➢ Store hashes of passwords instead of passwords (Michael Crinite)
- ➢ Email/Text message notifications from website (Joe Mecca)

Week 2 (3/12 - 3/18):

- ➢ Get and implement a chart-making API (Sean O'Connell)
- ➢ Create user profile page (Michael Moscariello)
- ➢ Track timeframes since last checked oil, entered mileage, etc. (Michael Moscariello)
- ➢ Implement means of displaying time since/until next maintenance (Michael Crinite)
- ➢ Create alert system to attach alerts to cars based on their information and display to the user on homepage and messages tab (Michael Crinite)
- ➢ Implement internal search engine (Joe Mecca)
- ➢ Create user preferences page which allows user to save settings (Andrew)

5. TECHNOLOGY STACK:

SparkJava (RESTful api), MySQL (Database), Amazon Web Services (AWS), Adobe PhoneGap (App Wrapper)

Web Development related tech:

- ➢ WebStorm IDE
- ➢ JavaScript
- ➢ JQuery
- ➢ AJAX
- ➢ Bootstrap
- ➢ Edmund's API (Car information)
- ➢ Google Maps API
- ➢ Google Chart API