Project Proposal: AutoTrack

Auto Maintenance Management Application

Lauren Malone

Regis University

MSSE 696 Software Engineering Practicum 2

Abstract

I often find myself racking my records and my calendar for the last time I had my car's oil changed. Sure, I have that little sticker that the mechanic put on my windshield with the next suggested service mileage; and that is great in theory, but sometimes it falls off, it's difficult to interpret the handwriting, or I forget it's there all together. And what about my last tire rotation? There are too many regular scheduled maintenance services to fit on that little sticker. On several occasions, my mechanic has recommended services that aren't necessary. AutoTrack will help track and manage scheduled auto services, and allow the user to review maintenance history for multiple cars. The user will have the ability to update the mileage as necessary.

Design Goals

Besides the application's core functionality, I am working on my design with the following non-functional requirements in mind:

1. Maintainability

Often times, rapid prototype development focuses on feature feasibility, but I'd like to take time focusing on best practices that keep the application maintainable so that it can evolve elegantly. Some of those practices will include limiting data duplication, using appropriate layering, and keeping my HTML, JS, and PHP separated. Business logic will live on the server side whenever possible.

2. Security

During development, I plan to implement a secure authentication service to maintain data confidentiality, integrity, and availability.

Envisioned Architecture

AutoTrack will be implemented as a full stack web application hosted on a LAMP stack. Initially, the development server will be stood up using VirtualBox; in the second half of Practicum, I plan to deploy the application to the cloud using AWS. PHP and MySQL database will interface utilizing Doctrine ORM. The client side will use the AngularJS framework, Material Design, and HTML. I plan to test the application logic by using Postman and to develop in PHPStorm IDE.

Future Evolution

While deciding on the initial design, I've considered several features that would be ideal for the application, but will likely not fit in the initial version, given the scope and time constraints of the class. AutoTrack will serve its purpose best as a mobile application in terms of convenience and accessibility. In the future, this expansion may be done, but it will be a simple web application during this prototyping stage. It will utilize a generic maintenance schedule, but it will be ideal to use the manufacturer suggested schedule for a given make, model, and year. In the future, this feature could be added. Additionally, it would be interesting to explore integration with a smart car so that mileage is tracked more accurately and precisely. Finally, new features concerning spending and fuel economy could be added as the application evolves.

Common Good Computing Considerations

Several auto tracking applications exist, but I've found that many are proprietary solutions, not user-friendly, or lack features such as offering cumulative history. AutoTrack will be an open source solution that can benefit its users. By maintaining the details of their required auto services, users can spend more time doing the things they care about and less time getting overcharged for unnecessary services. The scheduled reminders will support safer roads, and maintaining proof of regular services will help owners increase their car's resale value.

Additionally, making AutoTrack an open source project will enhance the computing community by enabling others to use and improve its services.