

Corey McCrea

Rene German

Database Management

## Final Project Writeup: Auto Maintenance Logger

### **The Problem**

This application seeks to fill the need of managing the maintenance any number of vehicles. Often service intervals vary between manufacturers and even car models. I sought to address this with a application and database that would consolidate management of invoices and other vehicle information into one place. I wanted to combat the status quo of throwing invoices into folders that seem to grow exponentially over time and develop something that could be used in the real world. That is also part of why I wanted to develop a polished GUI that would be intuitive for anyone to use. Currently my application tracks vehicle information, along with mechanic contact information and stores invoice information. This would give a user a centralized place to go to look up any information regarding their vehicles Want to know where a car was last serviced? How long it has been since a transmission service? Maybe the average cost of visiting a certain mechanic? All of this is possible with this application.

### **Related Work**

There are a number of desktop and mobile applications that seek to accomplish a similar goal. Notably Gas Buddy is an app that helps track gas prices nearby and seeks to find the best rate within a certain distance to you, it also performs minimal service tracking but nothing nearly all full-fledged as this application seeks to accomplish. There are also a few tools developed for use in car dealerships however these tend to be expensive and unoptimized for personal use. A

really excellent place for a competitor to enter this market would be with a cross-platform app that ran on mobile and desktop and synced all a given users data across platforms.

## **Framework**

This app was built in Python using PyQt for the frontend framework and MySQL for the backend. PyQt is a joy to work with due to its widget-based design process. It is extremely modular and adaptable for many uses. This application uses 11 widgets in total for its design. There are three tabs which correspond to each of my SQL views. These views populate a shared table used for displaying data. Each of these tabs have corresponding buttons for adding, updating, filtering, and deleting. In the vehicle tab users can filter by make and model. In the mechanic tab users can filter addresses containing a certain string. Invoices can be filtered by either mechanic, vehicle, or both. There is an interface python class built with all the necessary SQL commands which then PyQt widgets can signal to using the Slot/Signal framework built into Qt.

My database is comprised of five tables, invoice, mechanic, manufacturer, model, and vehicle. These tables hold values relevant to their titles. There are three views in the database, a vehicle view which combines vehicle, model, and manufacturer, an invoice view which hides some of the extraneous invoice columns, and a mechanic view which shows all the mechanic data. These views connected with the Qt objects made the overall development process exceptionally smooth. Filtering uses a subquery involving with the original data and a view. The model for the table in Qt is built on a pandas data frame which could be simplified to a Qt SQL Model. The generating reports functionality of my app outputs the total and average invoice amount for certain mechanics, manufacturers, and vehicles. These used joins across five tables

along with the use of group-by clauses. If an insert across multiple tables interrupted the database is designed to rollback all previous uncommitted changes.

## Results

Year	Model	Make	Color	Miles
2001	Sierra 2500 ...	GMC	BurlyWood	20000
2000	Savana 1500 ...	GMC	SlateBlue	1000
1994	Ram Wagon B250	Dodge	RosyBrown	1000
2016	Outlander Sport	Mitsubishi	Yellow	1000
1993	Navajo	MAZDA	LawnGreen	1000
2010	Terrain	GMC	ForestGreen	1000
2012	Sentra	Nissan	LightSalmon	1000
2006	H	INFINITI	CadetBlue	1000
2008	Savana 3500 ...	GMC	HoneyDew	1000
2011	Quattroporte	Maserati	Gold	1000
2002	Yukon XL 2500	GMC	OrangeRed	1000
1992	Yukon	GMC	Peru	1000
2015	Navigator	Lincoln	DodgerBlue	1000
2005	E250 Super Dut...	Ford	LightSkyBlue	1000

## Future Changes

There are several changes I would like to implement in the future with this application. Mainly, I would like to add certain codes for services in the description of an invoice so that I can display a service report for a given vehicle, printing out the last milage that every service was performed and notifying the user if any maintenance is needed. Another interesting idea that I had would be a small module that plugs into an OBD-II port on a car that updates the current vehicle milage every day in the MySQL Database. With these two potential features I feel that this app would be ready for real world usage and deployment.

## Schema Diagram

