Vehicle Maintenance Tracking Program

Jared Tomlinson
jet102@uakron.edu

Follow this and additional works at: https://ideaexchange.uakron.edu/honors_research_projects

Part of the Computer and Systems Architecture Commons, and the Data Storage Systems Commons

Please take a moment to share how this work helps you through this survey. Your feedback will be important as we plan further development of our repository.

Recommended Citation
https://ideaexchange.uakron.edu/honors_research_projects/1576

This Dissertation/Thesis is brought to you for free and open access by The Dr. Gary B. and Pamela S. Williams Honors College at IdeaExchange@UAkron, the institutional repository of The University of Akron in Akron, Ohio, USA. It has been accepted for inclusion in Williams Honors College, Honors Research Projects by an authorized administrator of IdeaExchange@UAkron. For more information, please contact mjon@uakron.edu, uapress@uakron.edu.
Senior Programming Project: Vehicle Maintenance Tracking Program

Jared Tomlinson
The University of Akron
Senior Programming Projects 2440:451:801
Dr. Eugene Mourzine
Abstract / Problem Description

Keeping track of vehicle maintenance can easily become a tedious process for anyone who owns multiple vehicles or businesses possessing large fleets. Blindly trusting auto shops or dealerships can also be an easy way to be taken advantage of and charged for unnecessary services and checks. All vehicles come from the dealership with a user manual that contains the maintenance schedule recommended by the manufacturer but tracking this maintenance and remembering to check on it regularly is something forgotten by many car owners. Neglecting this kind of maintenance can lead to issues varying from the inconvenience of poorly filtered air in the cabin, to premature engine wear, to the danger of missed drivetrain and suspension work.

To meet this need for both consumers and commercial clients, a tracking system should be created that allows users to easily track the maintenance of their vehicles. The system should be able to keep track of as many vehicles as needed and let the user know whenever they need to worry about maintenance.

System Capabilities

The system should be able to:

- Keep track of maintenance records and mileage for 1-100+ vehicles
- Allow users to easily update the mileage of each vehicle in the system
- Notify users of needed maintenance based on date and mileage changes
- Allow users to add, remove, or edit vehicles as needed
- Allow users to input and edit maintenance schedules for each vehicle in the system
• Users should be able to record performed maintenance

• Users can search for specific vehicles by make, model, year, color, and license plate

User Personas

• John and Sam Smith
  o Middle class parents
  o Own several vehicles between them and their children
  o Financially conscious
  o Like to plan ahead

• Chris Jenkins
  o Business owner
  o Operates a fleet of different vehicles
  o Values regularity in business procedures
  o Looking to invest to save money in the long-term

Business Benefits

After implementing the described system, businesses or private users should expect the following benefits.

• Improved reliability and uptime on vehicles due to proper maintenance routines

• Reduced long term maintenance costs from avoiding unneeded work and issues from missed maintenance

• More consistent maintenance scheduling due to regular checks and tracking

• Improved experience for drivers as vehicle features are better maintained
**Functional Requirements**

(In order of importance)

- Users can add vehicles with all their important information (make, model, year, color, license plate, etc.)
- A custom maintenance schedule is tied to each vehicle with various repairs and mileage intervals
- The system will track vehicle mileage and allow users to easily update its value
- System will display all vehicles and their current mileage.
- The program will notify users when mileage passes a maintenance interval
- Users can edit or remove vehicles from the tracker as needed
- Maintenance performed can be input and recorded on the maintenance schedule
- Users can search for individual or groups of vehicles by their information
- System will notify users of upcoming maintenance based on average mileage
- Maintenance schedule will account for early or surprise repairs
- Reports can be created for upcoming maintenance predictions based on all vehicles or a search
- Templates are available to easily create new vehicles and maintenance items
- Maintenance schedules can be saved and copied for vehicles with identical maintenance needs

**Technology Stack**

a. Windows Forms
b. C#
c. Entity Framework and Language Integrated Query
d. SQL Server Express Local DB

Sprint 1: (9-10 hours)

a) Create vehicle and maintenance item databases
b) Configured entity framework
c) Created home form and integrated vehicle table entities
d) Created add vehicle form and integrated with home form and vehicle table
e) Created details form based on maintenance table and selected vehicle
f) Created add maintenance form and integrated with details form and maintenance table

g)
Sprint 2: (10 hours)

a. Overhaul of UI design

b. Removed home form buttons in favor of datagrid interactions

c. Consolidated Details and AddVehicle forms

d. Vehicles can now be edited from the Details page

e. Added MaintenanceRecord table to database and Entity Framework model
Sprint 3: (14 hours)

a. Finished overhaul of UI and made it consistent across all forms with new dark mode format

b. Completed implementation of Maintenance record and its interactions with other objects and forms

c. Completed MaintenanceDetails form and all of its features including managing of maintenance records.

d. Redesigned warnings for vehicle maintenance so that it notifies for overdue with red highlighting and predicted due within 30 days with orange highlighting

e. Revised database and Entity Framework objects

f. [Image of vehicle maintenance tracking program interface]
Sprint 4: (12 hours)

a. Added unscheduled record table and updated Entity Framework

b. Added table for unscheduled records on vehicleDetails form and allowed for creation, editing, and deletion

c. Added filtering feature for main vehicle page based on status, mileage, make, model, and year

d. Added report to display Vehicle maintenance summary and cost/downtime totals based on all vehicles or filtered vehicles
### Vehicle Maintenance Tracking Program

#### Maintenance Schedule

<table>
<thead>
<tr>
<th>Title</th>
<th>Mileage Interval</th>
<th>Last Mileage</th>
<th>Next Mileage</th>
<th>Time Interval</th>
<th>Last Data Performed</th>
<th>Next Data Due</th>
<th>Average Cost</th>
<th>Average Downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil Change</td>
<td>10000</td>
<td>90000</td>
<td>13000</td>
<td>6 Months</td>
<td>5/29/2022</td>
<td>9/30/2022</td>
<td>$150.00</td>
<td>1 Hours</td>
</tr>
<tr>
<td>Timing Belt</td>
<td>100000</td>
<td>100000</td>
<td>20000</td>
<td>0 Months</td>
<td>10/30/2021</td>
<td>12/31/9999</td>
<td>$200.00</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

#### Non-Scheduled Repairs

<table>
<thead>
<tr>
<th>Maintenance Title</th>
<th>Mileage Performed</th>
<th>Date Performed</th>
<th>Cost</th>
<th>Downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door Replacement</td>
<td>70000</td>
<td>1/30/2022</td>
<td>$500.00</td>
<td>2 Days</td>
</tr>
</tbody>
</table>

#### Report

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>Maintenance</th>
<th>Downtime</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>H643534</td>
<td>Oil Change</td>
<td>2 Hours</td>
<td>$50.00</td>
</tr>
<tr>
<td></td>
<td>Oil Change</td>
<td>1 Hours</td>
<td>$40.00</td>
</tr>
<tr>
<td>Subtotal</td>
<td>3 Hours</td>
<td>$90.00</td>
<td></td>
</tr>
<tr>
<td>ABC123</td>
<td>Oil Change</td>
<td>1 Hours</td>
<td>$40.00</td>
</tr>
<tr>
<td>Subtotal</td>
<td>1 Hours</td>
<td>$40.00</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4 Hours</td>
<td>$130.00</td>
<td></td>
</tr>
</tbody>
</table>
VEHICLE MAINTENANCE TRACKING PROGRAM
Sprint 5: (16 hours)

a. Further cleanup of UI and added icon

b. Report feature now displays predicted and overdue maintenance items when a future date is used

c. Added vehicle and maintenance templates tables and integrated Entity Framework

d. Added template management pages for creation, edition, deleting, and linking templates

e. Template dropdown is available when creating a new vehicle or maintenance item

f. Code cleanup and refactoring
### VEHICLE MAINTENANCE TRACKING PROGRAM

#### Maintenance Templates:

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Mileage Interval</th>
<th>Time Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil Change</td>
<td>Oil Change</td>
<td>3000</td>
<td></td>
</tr>
</tbody>
</table>

#### Vehicle Template:

- **Name:** [Example Value]
- **Make:** [Example Value]
- **Model:** [Example Value]
- **Year:** 2006

#### Unlinked Maintenance Templates:

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Mileage Interval</th>
<th>Time Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil Change</td>
<td>Oil Change</td>
<td>3000</td>
<td></td>
</tr>
</tbody>
</table>
Backlog (Unfinished Requirements)

a. Maintenance schedules can be saved and copied for vehicles with identical maintenance needs

b. Importing and exporting of database information using csv files / Excel

c. Further user interface improvements

Conclusions

The final version of the program satisfies almost all of the original functional requirements that I had set out for the project, and I believe that it effectively solves the problem that I set out to remedy. The few missing features would allow for better ease of use for customers, but they are not integral to the program’s main functions. That being said, no program is ever truly complete, and improvements and bug fixes are always a possibility for the future.

The technology stack that I chose for this project worked well for the logic and technical requirements of the program but ended up limiting my user interface and design options. The combination of Entity Framework and C# made handling database integration simple and straightforward which was crucial for this project and all of its tables. On the other hand, I believe that Windows Forms could have been replaced with a better front end such as an MVC web page project. This would have allowed for a simpler navigation system and more customization for interface design.

Even with the limitations of Windows Forms, this aspect of the project is where I learned the most. I had never dealt with integrating database information into windows forms.
before the start of this program, but this became one of the most frequently used features in my final product. The limitations in styling also led me to pushing the styling features available as far as possible to improve the end user experience. Entity Framework and Language Integrated Query (LINQ) is another area where I learned a lot. Although I have used these features within C# in the past, I have not used them with this many interconnected tables or when passing database information between windows forms. Learning how to properly filter down table rows in order to find and display the correct content into grids and forms was fundamental to every page in the program.

The technical skills gained from this project are not the only things that I learned, however. I also gained experience with time management, goal setting, and agile development processes. As my project progressed through the multiple development sprints, I slowly improved my ability to predict the difficulty and time needed to complete upcoming requirements. This, along with better managing my development time, made my sprints more productive the later in the semester that they occurred. These skills are the most valuable take away from this project and will be extremely useful in a commercial software development environment.