Summary

“Development of a fleet management system”

Monika Mariova Spasova, fac. № 25600, monikaspasova1@gmail.com

Asst. prof. Dr. Galia Novakova, Department of Computing Systems, Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”, g.novak@fmi.uni-sofia.bg

Abstract

In recent years fleet management has turned into a multi-billion industry, which continues to grow and is gaining significant strategic importance in a world of changing mobility. In Europe, by far the largest market for fleet management globally, nearly two out of three new cars are sold to the corporate channel. In consequence, the majority of these vehicles are registered as company cars, whose maintenance companies aim to optimize. [1] A high percentage of the existing fleet management systems find difficulties with the vehicle’s integration. The application, a subject of the master thesis, is aimed at their removal.

Keywords: Telematics, AEM/AEMP standard, Fleet management, Optimizations

Introduction

Relevance of the problem and motivation

On the market there is a variety of fleet management software solutions. However, a lot of them are not connected to telematics equipment. The majority of the ones, having such an integration, are targeted at large corporations and are offered at high prices. Moreover, they are used to work with telematics equipment of a narrow set of manufacturers. The existence of such dependency between software and hardware significantly limits the possibility for integration with another software when the telematics devices are already mounted in the vehicles and vice versa.

These facts motivate the idea for the development of a fleet management system, targeting the small and medium-sized business. It encompasses basic functionalities regarding the scale of this category of companies like the management of vehicles, drivers and services. The system is going to be integrated with the universal AEM/AEMP standard for a common data format.

Goal

The goal of the present thesis is the development of a user-friendly fleet management system, compatible with telematics equipment with a different origin.

Benefits

- Systematized and centralized management of the fleet management components data
- Tracking of vehicles characteristics in real time or for a past period
- Better control over the drivers and vehicles
- Prevention of trials for misappropriations with corporate vehicles
- Timely maintenance
- Reduction of fuel expenses
- Reduction of maintenance expenses

Functional characteristics

The access to the system is performed through a registration and login form.
The application consists of the following modules:

1) **Companies** – Enables creation, editing, deletion and a preview of a company, as well as providing access to a company. The registered user can act as an administrator or a guest of a company. Depending on this, he has different powers over it in the system.
2) **Drivers and vehicles** – These modules manage the information about the drivers and, respectively, the vehicles in a company. The modules allow CRUD (Create, Update, Delete) operations with drivers and vehicles, as well as the ability to connect a vehicle to a driver.
3) **Services** – This module manages the vehicles services information. The system provides functionality for the creation of tasks, based on time or mileage.

   If the task is based on time, the user should enter:

   - the period for the execution of regular tasks;
   - how much time before the date of the execution a reminder needs to be sent;

   When a task is created, the time for the next reminder and execution are calculated, based on the entered data.

   If the task is based on mileage, the user enters:

   - the mileage for the execution of regular tasks;
   - how many kilometers before the mileage of the execution a reminder needs to be sent;

When a task, based on the mileage, is saved, the mileage for the next execution and reminder are calculated. The sending of emails about the following and overdue tasks are based on these calculations.

The user can mark a task as done. Then the next date or mileage for reminder and execution are calculated, based on the entered rules.

4) **Reports** – The module allows preview of mileage and fuel level information in real time and one week back (with chart).

Technical realization

1) **Generation of telematics data** – This functionality is created for the purpose of the master thesis due to the lack of real telematics equipment. The generation of telematics data is done through a *SeedTelematicsJob* task, started every 60 minutes. When the task is executed, for each vehicle in the database the mileage and the fuel level are updated. Algorithms are created to generate valid mileage (constantly increasing) and fuel level values.
2) **Usage of telematics equipment** – In reality, the application will be used with data from vehicles telematics equipment. For this purpose, the system will make requests to the telematics provider API-s every 20 seconds, using vehicles VIN numbers. In fact, the main difference with the current version of the system will be how the system receives the mileage and fuel data – in the current version through the data generation class methods, in the future through requests to the API-s of the telematics equipment manufacturers.
3) **Sending emails for upcoming and overdue tasks –** Emails are sent through daily tasks – *SendMileageReminderEmailJob, SendTimeReminderEmailJob, SendMileageOverdueEmailJob, SendTimeOverdueEmailJob*. They compare the current date (mileage) with the stored data for the next execution or reminder. Based on this comparison, notifications of upcoming and overdue tasks are sent.
4) **Modules for companies, drivers, vehicles and tasks –** In the server side they are fully realized using the basic levels of architecture. They include *Get*, *Post, Put* and *Delete* methods involved in performing the relevant operations. The *DataAccessService* level through the *Data (EntityFramework)* models performs the database operations.

Architecture and used technologies

The system is of type REST API (*Representational State Transfer API*). It consists of two main sides – client and server. The architecture of the server side is composed of the following levels: *WebApi*, *Business*, *DataAcces, Data (Entityframework)*. The *Code first* approach is used to create the SQL database. The API is implemented with the *ASP.Net Web API 2*. The *Quartz* library is used for the regular tasks execution (generation of telematics data and sending emails). *Alt.js* is the library, used for the client.

Conclusion

Generally, fleet management is based on telematics data. Although the usage of telematics in companies with small fleets is not very common, according to a *Financial Times* article in the next five to seven years telematics equipment will become an integral part of all vehicles. [2] In such a situation, the need for easy integration between different types of software and hardware will significantly increase. In order to ensure compatibility, systems, integrated with the AEM/AEMP standard, will be needed. A prototype of such a system is created within the framework of the thesis.

Literature

[1] Deloitte, “Fleet management in Europe”, July 2017, https://www2.deloitte.com/content/dam/Deloitte/cz/Documents/consumer-and-industrial/cz-fleet-management-in-europe.pdf

[2] “Telematics Is Revolutionising Fleet Management.” Financial Times, 18 Apr. 2016, www.ft.com/content/ca557812-c03a-11e5-9fdb-87b8d15baec2?siteedition=intl#axzz4JVuwLBje.