System Documentation

SYSA21: IS-Projekt | 2023/2024

Group: **Arbetsgrupp 10**
GitHub Team: **Javengers**
GitHub Repository: https://github.com/lu-informatics/ht23-sysa21-isp-ht23-sysa21-javengers

980629-7875	David Allan	dallan5
030802-6715	Hugo Forsgren	HugoForsgren
940213-7443	Saga Jenninger	sagajenninger
040608-9243	Izzy Tiger	IzzyRTiger
020313-9134	Johan Wahlgren	JohanWahlgren

Overview

Your systems documentation must contain a succinct overview of your system's purpose, detailing how your application supports the organizational goals, business processes, and tasks as outlined in the case description.

Our system addresses the specific needs of Viking Express for quality and regulatory compliance by closely monitoring and managing their fleet of vehicles. The application aims to provide real-time information on vehicle locations, availability, and on-demand access to comprehensive maintenance and service history.

The system manages vehicles with unique Vehicle Identification Numbers (VINs) and associated names, categorizing them into allowed types: Large Truck, Medium Truck, and Van. It is capable of displaying information on all vehicles, including their types, current locations, and capacities. Additionally, the system presents maintenance schedules, service histories, and details on workshops, whether internal or external.

The graphical user interface facilitates the creation, reading, updating, and deletion of information related to vehicles, workshops, service history, and maintenance schedules. It offers functionality to display complete service histories for specific vehicles, service entries for particular workshops, and enables the addition and removal of service activities.

The system enforces business rules such as non-negativity of a vehicle's capacity, preservation of the uniqueness of the VIN, and automatic generation of VINs. It issues a warning if the total cost of servicing a vehicle exceeds $100,000 and implements constraints like limiting the number of parts replaced and decommissioning a vehicle with over 100 replaced parts. The system calculates the average cost of vehicle maintenance, displays information on the most expensive maintenance job and workshop, and lists workshops where a specific vehicle has been serviced.

Finally, the software is intended to be designed with an intuitive user dialogue and graphical user interface, utilizing appropriate elements for user tasks. It includes error handling with user-friendly messages and presents data using tables for easy comprehension.

System architecture

You must also include models and diagrams documenting your system architecture. All diagrams must be drawn using modeling tools. Hand-drawn diagrams, even if scanned or otherwise digitized, will not be accepted.
• Class diagram showing the structure of your Java application. Exclude setter- and getter methods, action listeners, and instance variables tied to graphical user interface elements.

(KLISTRA IN DET SLUTGILTIGA KLASSDIAGRAMMET FÖR PROGRAMMERINGSUPPGIFTEN)

Software

Software used in this project:

-Visual Studio Code
-GitHub
-SceneBuilder
-Diagrams.net (draw.io)
-ChatGPT
-Google docs
-Google sheets
-Messenger (Facebook)

Libraries

For this project, no third-party libraries have been used.

Known issues

Known issues: Detail any known bugs, limitations, or concerns regarding the current iteration of your system.

Known bugs

There are currently no known bugs in the system. (?)

Limitations

-If the user wants to remove a service activity for a workshop in either the maintenance schedule or the service history for that workshop, all of the service activities will be deleted.

-When right-clicking on a vehicle, the option "Show Total Cost" appears. It may be somewhat unclear to the user that we are referring to the "Show Total Cost of Service Activities for the specific vehicle." However, it has been abbreviated to "Show Total Cost" due to the impractical length of a more detailed description

Concerns

Recommendations

Suggested improvements or additions for future versions

-Implement a notification system for upcoming maintenance schedules.
-Implement a feature to generate reports summarizing the overall maintenance costs for specific vehicle types.
-Allow users to compare maintenance costs between different workshops.
-Ensure that a vehicle cannot be assigned to more than one workshop for maintenance at the same time.
-Set a restriction on the frequency of service activities for each vehicle to prevent excessive maintenance.
-Implement search functionality for vehicles based on their names or types.
-After removing a vehicle, there could be an option to restore the deleted vehicle.