RESEARCH ON COMPUTER APPLICATIONS FOR MANAGING FREIGHT SHIPMENTS IN A FACTORY

COSTACHE Filofteia Bianca
Faculty of Industrial Engineering and Robotics, Specialty: Applied Informatics in Industrial Engineering, Year of study: 4, e-mail: filofteia.costache@stud.fiir.upb.ro

ABSTRACT: This document presents a web application for managing logistics in freight transportation. The application provides a comprehensive solution for visualizing and managing logistical events associated with cargo transport. The users are carriers and administrators. Carriers can access a main page displaying the list of transports and an interactive map for optimal routing. Administrators can view and manage carriers, transports, goods, and generate reports. The application utilizes technologies such as React for the user interface, Node.js and Express.js for the server, and SQLite for the database. It differentiates itself through automatic transport scheduling and internal carrier management. Planned future functionalities include integrating a transport status monitoring system and developing carrier-specific pages and an optimal routing algorithm.

KEY WORDS: merchandise, transport, database, system, user

1. Introduction

The areas chosen for research were: (1) software development for web applications and (2) logistics.

The aim of the research was to digitise, automate and improve the quality and safety of the economic processes that take place during the logistical transport of goods of a company producing a certain type/types of products.

The way in which these objectives have been achieved is the construction of a complete web-based solution for the visualisation and management of logistical events associated with freight transport. The solution is structured as a web portal with two main interfaces: the interface viewed by a transporter employed within the company and the administrative interface of the administrator.

2. Application Flows

The proposed application presents 2 main flows: carrier flow and administrator flow.

2.1. Transporter Flow

Upon initial access to the application, the carrier is presented with a form of authentication, which is done by username and password. After authentication, he/she is redirected to the main page of the application. This page contains a list of all the transports associated with the employee for that day, in the order generated by the timetable algorithm, as well as an interactive map highlighting the optimal route to the next destination. The map will be updated according to the current transport to be performed.

In addition to the main page, the carrier also has a profile page, where they can view and modify their personal details such as email address, phone number, password, etc.

2.2. Administrator Flow

The administrator can perform the following operations within the application:
1. View carriers.
2. View completed, ongoing and future shipments.
3. View and add goods to be transported.
4. View, add and associate to the driver of the transport vehicle.
5. Automatic triggering of the timetable algorithm.
6. Generate reports with details of company activity.

The administrator account will be unique and recognized by the application, with the modification of the interface corresponding to the logged in user.

3. Architecture and Components

3.1. Functional Entities

There are 5 main entities in this system, namely Carrier, Transport, Administrator, Freight and Carrier [1]. The roles of these entities are described as follows :

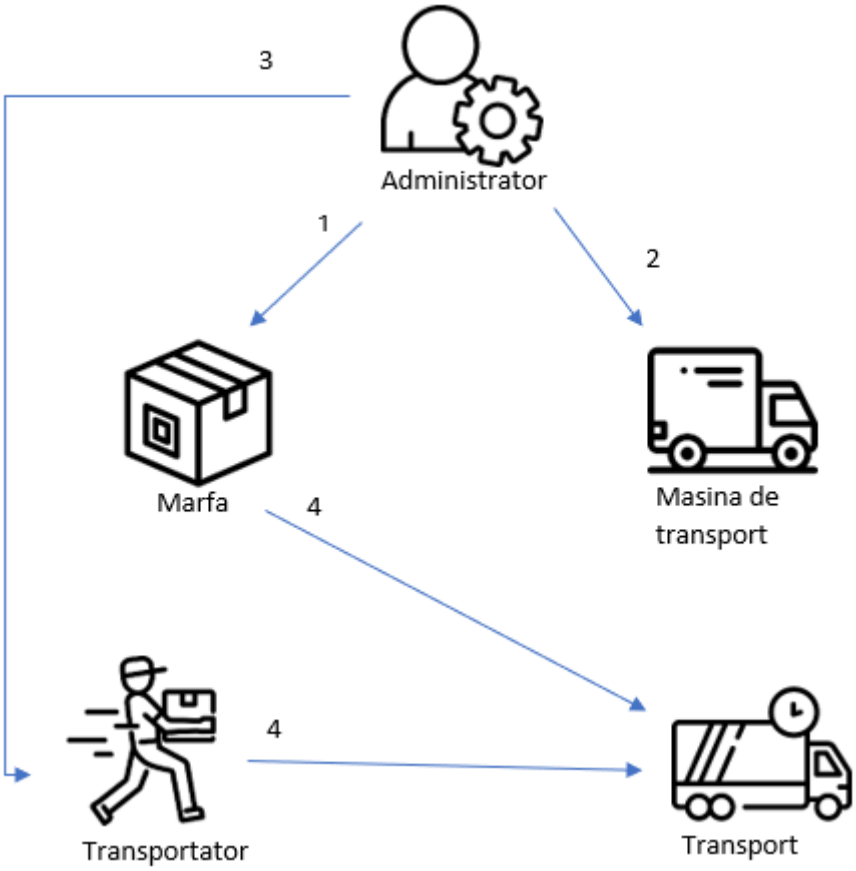

Fig. 1. System Architecture Diagram

Step 1: The administrator enters the goods to be delivered in the next period into the system.

Step 2: The administrator enters into the system the transport vehicles available within the company. This step is carried out when adopting the solution and when the company changes its fleet of vehicles or purchases new ones.

Step 3: The administrator triggers the scheduling algorithm, which will assign the day's transports to each individual haulier.

Step 4: The carrier views the shipments to be made in the web interface, along with details of the cargo and the location it is due to arrive at.

3.2. Designing Database

The database is composed of the following entities: Carrier, TransportingCar, Merchandise, Transport (Fig. 2).

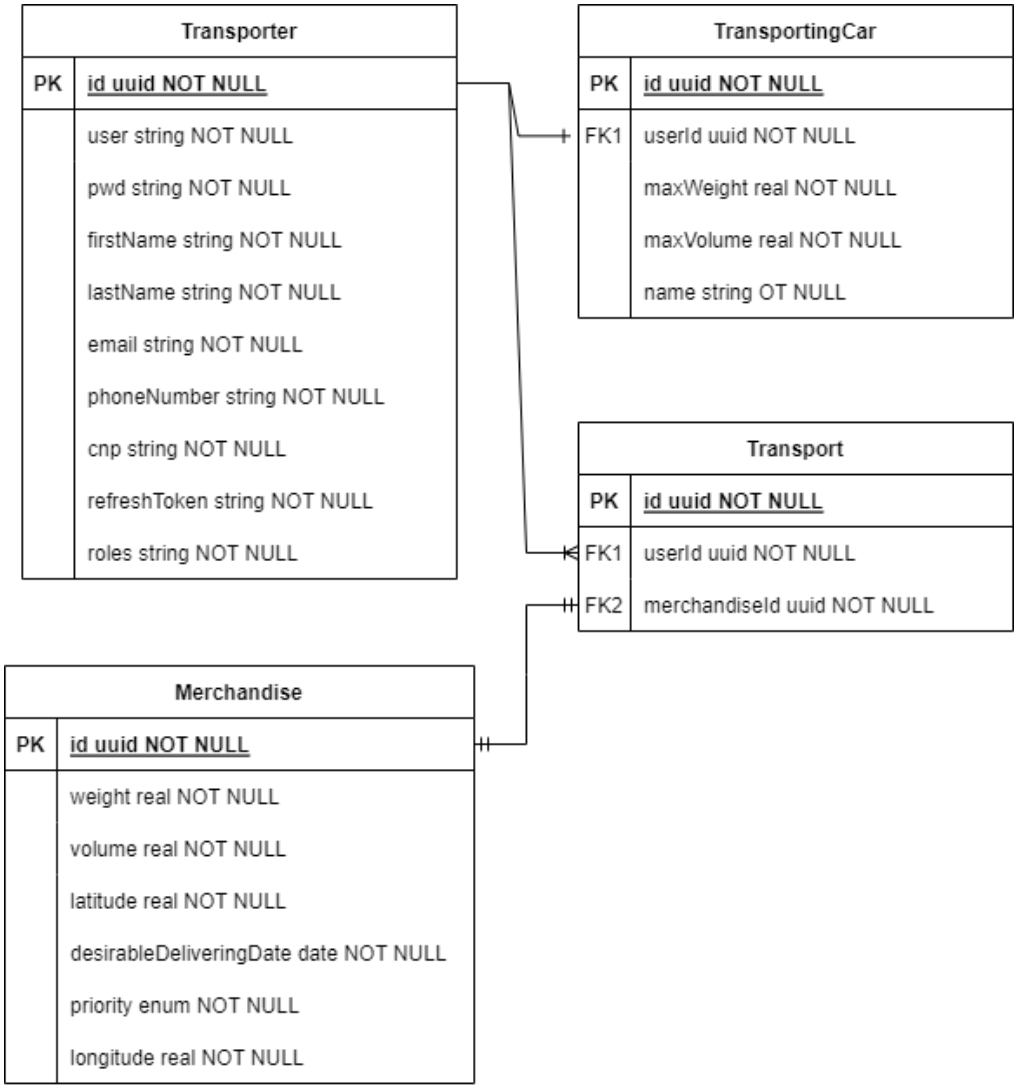

Fig. 2. Database Design

Entity relationships [8]:

1. Between Transporter and TransportingCar 1 to 1 relationship optional. A transporter can have only one car, and a car can be driven by only one transporter.

2. Between Transporter and TransportingCar relationship 1 to many. A transporter may have several transports to carry out in one day, and one transport is carried out by one transporter.

3. Between Transport and Merchandise 1 to 1 relationship mandatory. A transport has only one commodity, a commodity is associated with only one transport.

4. Market Analysis

Startup "Cargo Buddy" is a prominent competitor in the freight market [2]. At its core is a mobile app that allows customers to ship goods domestically and internationally. Anyone can sign up for a free account, manually entering the goods they want to deliver and the date they want them to arrive at their destination.

Carriers listed within the app are selected according to the customer's criteria to pick up and transport the goods.

One of the major benefits of the app is the ability for customers to monitor the real-time status and location of their shipments.

The solution presented in this document differs from Cargo Buddy in that it is oriented towards a single customer's internal environment. In addition to the functions shared with Cargo Buddy, it is able to manage the company's carriers, automatically assigning them a delivery schedule according to customer requirements. This extends the application to be used both by the company at macro level and by carriers at micro level, who can efficiently organise their working day according to the application's indications.

Another important competitor on the Romanian market is the xTrack TMS application, offered by the Romanian company Axes Software [3].

This application is intended for an ecosystem composed of several companies that can choose to organise their transports through this application.

It is composed, like the solution presented in this paper, of two main flows: the administrative flow for the distribution companies registered in the application and the carriers' flow.

A benefit of this application is the performance of "What If" analyses that include the following:

1. Cost estimation in case of a new distribution contract.
2. Identification of optimal locations for logistics centres.
3. Organisation of internal transport between warehouses, if the company has more than one.

5. Technologies

Technologies used in building the application :

1. User interface
    a. React - JavaScript framework used in building SPA (single-page applications) [7].
    b. HTML and CSS - fundamental website technologies [6].
    c. Bootstrap - a CSS library that makes it easier to create designs and lay out elements.
    d. TomTom Maps - library used for building customizable maps.

2. Server
    a. Node.js - JavaScript engine that allows JS code to be executed outside the browser [4] [5].
    b. Express.js - Node.js framework with which the interface routing table and API to the database are built [4] [5].
    c. Sequelizer - ORM (Object Relational Mapping) library that makes the connection between an external database and the web application.
    d. SQLite - minimalist database that is easy to integrate into web applications.
    e. Nodemailer - Node.js library that facilitates the sending of emails by the application.

3. Schedule algorithm - algorithm developed from scratch that retrieves goods with all their information (destination, quantity, weight, delivery date, indicated time of delivery, type of goods, etc.) from the database and builds a schedule for their delivery based on the carrier's work schedule, type of car driven, goods information.

4. Optimal route calculation algorithm - algorithm that will determine the optimal route to the destination based on the current position.

6. Current State

At the moment, the application is able to handle most of the flows associated with the administrator. The single user with administrator role can perform daily tasks such as viewing carriers, adding vehicles and associating them with carriers, adding freight, viewing shipments and triggering the shipment association algorithm.

7. Conclusions

The main personal contributions are :

1. Design and implementation of the application infrastructure - database design, building the API for communication between the database and the visual interface, making the link between the database and JavaScript, defining the application routing protocol.

2. Researching an algorithm for making a transport schedule, based on various criteria and business needs.

Functionalities to be implemented in the next period:

1. Integration of the status system - view status of carriers (delivery in progress, in warehouse, delivery made), status of transports (made, not made), status of goods (delivered, in warehouse).

2. Administrative page - view graphs and reports detailing the company's logistics activity (shipments made, delays, main customers)

3. Carrier flow - creation of pages associated with the carrier (home page, profile page).

4. Algorithm for determining the optimal route and highlighting it on the map.

8. References

[1]. *Ce este o diagramă de arhitectură de sistem pentru aplicații web?* (2020, July 29). From Vavavoom: https://vavavoomproductions.com/ro/ce-este-o-diagrama-de-arhitectura-de-sistem-pentru-aplicatii-web/

[2]. Coșman, O. (2021, March 2). *Cargo Buddy, un "Uber" românesc pentru transportul de mărfuri*. From start-up: https://start-up.ro/cargo-buddy-aplicatie-transport-marfuri/

[3]. *Despre xTrack TMS*. (without year). From softlead: https://softlead.ro/aplicatii-software/xtrack-tms.html

[4]. Schmedtmann, J. (without year). *Node.js, Express, MongoDB & More: The Complete Bootcamp 2022*.

[5]. On November 1, 2021, From Udemy: https://www.udemy.com/course/nodejs-express-mongodb-bootcamp/

[6]. Schmedtmann, J. (n.d.). *The Complete Javascript Course*. Retrieved June 1, 2021, from Udemy: https://www.udemy.com/course/the-complete-javascript-course/

[7]. Schwarzmüller, M. (2023, April). *React - The Complete Guide (incl Hooks, React Router, Redux)*. From udemy: https://www.udemy.com/course/react-the-complete-guide-incl-redux/?kw=react&src=sac#instructor-2

[8]. Team, M. 3. (2019, September 24). *Ghidul simplu pentru diagrame UML și modelarea bazelor de date*. From microsoft: https://www.microsoft.com/ro-ro/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling

Acknowledgement

I would like to thank to Conf.dr.eng. Ovidiu Dorin ALUPEI COJOCARIU for the support and guidance in completing this research.