Design and development of a dispatch board for public transport operators based on web technologies (PE43)

Capstone Project Final Report

António Ferreira
Cristiano Rocha
José Ferreira
Pedro Magalhães

Bachelor of Informatics and Computer Engineering

U.Porto Tutor: Teresa Galvão
Proponent: Thiago Sobral

Date: 2024-06-21

Contents

1 Introduction

This report aims to provide an overview of our project that involved the implementation of a system for a dispatch board for public transport operators.

1.1 Background

The project was carried out within the OPT facilities (Otimização e Planeamento de Transportes S.A). We were accompanied by our tutor from OPT, Thiago Sobral, who guided us through the entire work and helped us with everything we needed.

1.2 Objectives and Expected Results

The main motivation behind this project was the need to upgrade the already existing software used to display the dispatch board to a programming language that could be more easily managed, maintaining the existing features, and database integration. The existing software was considered outdated and hard to deploy since it ran natively on the devices, every update and bugfix required a manual software update of all the machines that needed it. The development of this dispatch board was based on an existing program, our job was to upgrade it by:

- adapting it to a web-based language, so that it is easier to expand and debug, taking advantage of the best and most recent tools of web-dev;
- adding customizations and "quality of life" features to improve the overall usability of the product.

1.3 Report Structure

This report will have the following structure:

- **Introduction:** Brief description of the initial background, motivation and context behind the project and the expected results.
- **Methodology and Development Process:** Description of the methodologies and main activities that were carried out in this project, including the methodology that was used, the intervenients and their roles/responsibilities, and the main activities that were developed during the project.
- **Solution Development:** The requirements and restrictions of the final product and the architecture and technologies used.
- **Conclusion:** Conclusion of the report with a summary of what we as a group achieved with this project and what we learned.

2 Methodology and Development Process

2.1 Methodology

To ensure consistent progress and receive regular feedback, our group established a weekly meeting schedule with our OPT tutor. These meetings allowed us to review our work, discuss any challenges, and make necessary adjustments based on the feedback received. For version control and collaboration, we utilized GitHub. We created a repository where we committed our code, which greatly enhanced our communication and organization. This platform allowed us to track changes, manage different versions of our project, and work simultaneously without conflicts. Our online meetings were conducted through a Discord group we created specifically for this project. We utilized both text and voice chat features to communicate in real-time while working on the project. Additionally, we included our OPT tutor in the Discord group so he could monitor our progress and provide feedback directly within this collaborative environment. For coding, we chose Visual Studio Code as our primary editor. This tool offered an array of extensions and features that facilitated our development process, making it easier to write, debug, and collaborate on our code efficiently.

2.2 Stakeholders and roles

Several people were involved in the project, each with specific roles and responsibilities. The stakeholders and their roles are as follows:

- Project Team(4 members):
  - António Ferreira: Frontend Developer
  - Cristiano Rocha: Frontend Developer
  - José Ferreira: Frontend Developer
  - Pedro Magalhães: Backend Developer
- Project Coordinators:
  - Professor and advisor Thiago Sobral from OPT: Responsible for advising and mentoring us, providing expertise and guidance.
  - Professor and supervisor Maria Teresa Galvão Dias from FEUP: Oversees the project and provides guidance and feedback.
  - Professor and Director of Capstone Project (Projeto Integrador) Nuno Flores: Conductor of Projeto Integrador.
- Project Users:
  - Transport Operators: The primary users of the dispatch board system.

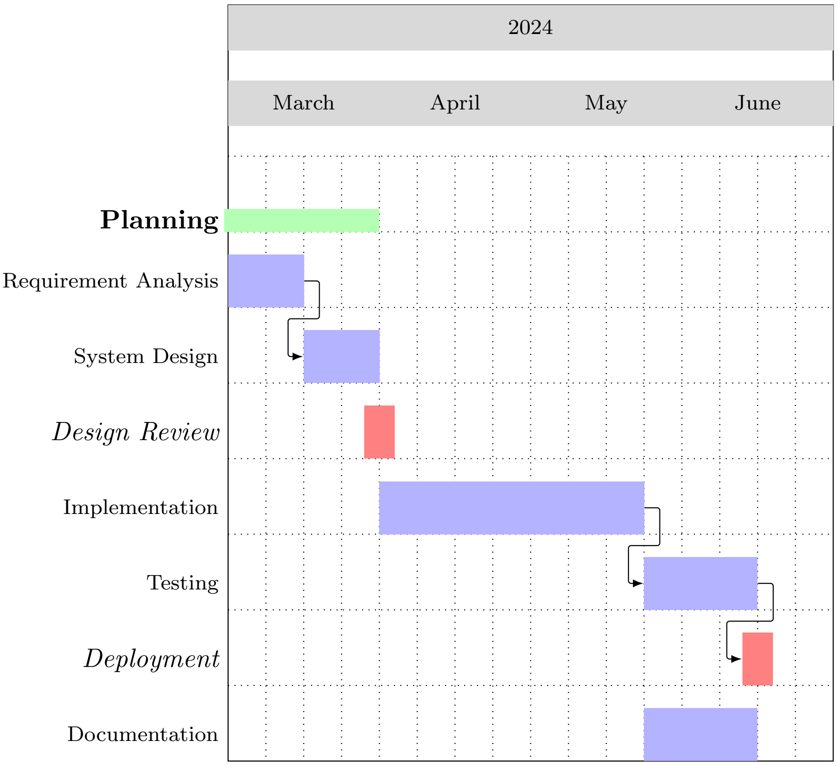
2.3 Activities Developed

During the project's duration, several activities took place, such as planning, coding and team meetings.

- **Planning:**
  - The initial phase involved determining the technologies to be used. In a meeting with our OPT coordinator, we decided on Node.js and Prisma for the backend, and React with TypeScript for the frontend.
  - We also discussed the project's requirements and constraints, such as the need for real-time updates and the importance of maintaining the existing features.
  - After the initial planning, we began the requirement analysis, which involved understanding the existing system and identifying the features that needed to be implemented in the new dispatch board. We consulted with our OPT coordinator to ensure that the new system met the required specifications.

- **Development:**
  - The development phase began with setting up the project structure and creating the initial files. We were provided a database from one of the existing dispatch board systems, which we used to test the backend. We were also provided the queries used to retrieve the data from the database.
  - We then proceeded to implement the backend, focusing on the database connection and API endpoints. We used Prisma to interact with the database and Next.js to create the API routes.
  - Another important aspect was the implementation of a way to customize the dispatch board, allowing users to change the colors and layout and the logo according to their preferences. We decided to use a JSON file to store these customizations, so that the users could export and import their settings to other devices.
  - The frontend development followed, with the creation of the user interface and integration with the backend. We used React with TypeScript to build the frontend, focusing on the real-time updates and customization features. There was a focus on creating a simple and intuitive interface that would be easy to use by the transport operators.
- **Testing:**
  - We conducted several rounds of testing to ensure that the system was functioning correctly and that all features were working as expected. We tested the real-time updates and customization options to verify that they were working properly.
  - We also conducted user testing with the transport operators to gather feedback on the system's usability and identify any issues that needed to be addressed.

Here's a Gantt chart showing the project's timeline, as it can be seen, the project was divided into four main phases: Planning, Requirement Analysis, Implementation, and Testing. After the initial planning, we began the requirement analysis, which involved understanding the existing system and identifying the features that needed to be implemented in the new dispatch board. We consulted with our OPT coordinator to ensure that the new system met the required specifications. The development phase began with setting up the project structure and creating the initial files. We were provided a database from one of the existing dispatch board systems, which we used to test the backend. We were also provided the queries used to retrieve the data from the database. We then proceeded to implement the backend, focusing on the database connection and API endpoints. We used Prisma to interact with the database and Next.js to create the API routes. Another important aspect was the implementation of a way to customize the dispatch board, allowing users to change the colors and layout and the logo according to their preferences. We decided to store the user settings on the local storage, so that the users could export and import their settings to other devices.



3 Solution Development

3.1 Requirements

The main requirements for the dispatch board system were as follows:

- Functional Requirements:
  - **Customization:** The system should allow users to customize the dispatch board according to their preferences, such as changing the colors, logo and layout(display the columns that they consider useful and hide the ones that are considered useless for that company, also switch the order of the columns).
  - **User-friendly interface:** The system should have a simple and intuitive interface that is easy to use by the transport operators.
  - **Database integration:** The system should be able to connect to a database to retrieve the necessary information about the buses and schedules.
  - **Modular architecture:** The system should be modular, allowing for easy expansion and maintenance. It should also be possible for many transportation companies to use the same system, with each one having its own customizations and schedules.
- Non-functional Requirements:
  - **Reliability:** The system should be reliable, with real-time updates ensuring that the transport operators have access to the most up-to-date information.
  - **Usability:** The system should be easy to use, with a user-friendly interface that is intuitive and accessible to all transport operators. This is especially important for operators who may not be familiar with technology. The system should also be in Portuguese, as most of the transport operators don't speak English.

  - **Performance:** The system should be fast and responsive, with minimal lag or delay when loading the schedules. This is crucial for transport operators who need to access the information quickly and efficiently.
  - **Scalability:** The system should be scalable, able to handle a large number of users and schedules without compromising performance. This is important as the system may be used by multiple transportation companies simultaneously.
  - **Lightweight:** The system should be lightweight, with minimal resource requirements. This is important as the system may be used on older devices with limited processing power or lightweight raspberry pi-like devices.

- Constraints:
  - **Web-app based:** The system should tings from a JSON file.be developed as a web application, so that it can be accessed from any device with a web browser.
  - **Keep the existing features:** The system should maintain the existing features of the dispatch board, such as displaying the 'buses/trains' schedules.
  - **Real-time updates:** The system should provide real-time updates, ensuring that the transport operators have access to the most up-to-date information.

- Assumptions:
  - The transport operators have access to a device with a web browser, such as a computer or tablet.
  - The transport operators have a basic understanding of technology and can navigate a web application.

- Dependencies:
  - The system depends on the database to retrieve the necessary information about the schedules.

3.2 Architecture and Technologies

3.2.1 Technology choice and reasoning

Before embarking on the development of the dispatch board system, careful consideration was given to defining the optimal architecture and technologies. The selection of these components was crucial to ensure scalability, reliability, and ease of maintenance, while aligning with the project requirements and existing systems. With the freedom to choose the technologies, extensive research and discussions were conducted to make informed decisions. After thorough evaluation, the following technologies were deemed most suitable for the project:

- **Web Framework:**
  - **Description:** Next.js is a popular web framework for building modern and responsive user interfaces. It is built on top of React, a JavaScript library for building user interfaces, and provides additional features and optimizations specifically designed for server-side rendering and static site generation.
  - **Reasoning:** Next.js was chosen for its flexibility in rendering strategies, performance optimizations, and ease of development. It offers a modern and fast user interface, making it a perfect fit for this project.

- **Frontend:**
  - **Technology:** React with TypeScript was chosen for the frontend development. TypeScript is a superset of JavaScript that adds static typing to the language, providing enhanced code quality and improved developer productivity. React is a powerful library for building user interfaces, offering a component-based architecture that simplifies the development process.
  - **Reasoning:** React's component-based architecture simplifies the development process and enhances code quality. TypeScript adds static typing to JavaScript, providing improved code quality and developer productivity.
- **Backend:**
  - **Description:** We utilized an existing SQL Server database for the project, which contained the necessary information about the buses and schedules.
  - **Reasoning:** The SQL Server database provided by OPT was a suitable choice for the project as it already contained the necessary information about the buses and schedules. Leveraging this existing database allowed us to focus on developing the frontend and backend components without the need to create a new database from scratch.
- **ORM (Object-Relational Mapping):**
  - **Description:** Prisma is an advanced ORM that simplifies database interactions by providing a type-safe and intuitive API. It supports multiple databases, including PostgreSQL, SQL Server, MySQL, and SQLite, making it a versatile and efficient choice for the project.
  - **Reasoning:** Prisma was selected for simplified database interactions and data manipulation. It is easy to utilize in Next.js and offers a seamless solution for the project's data handling needs.

3.2.2 Architecture

The architecture of the dispatch board system was designed to be easy to maintain, easy to use, and efficient. It consists of three main components: the frontend, backend, and database. These components work together to provide a seamless user experience for transport operators, ensuring real-time updates and customization options. To represent the architecture, we created a layer diagram that illustrates the interaction between the components.

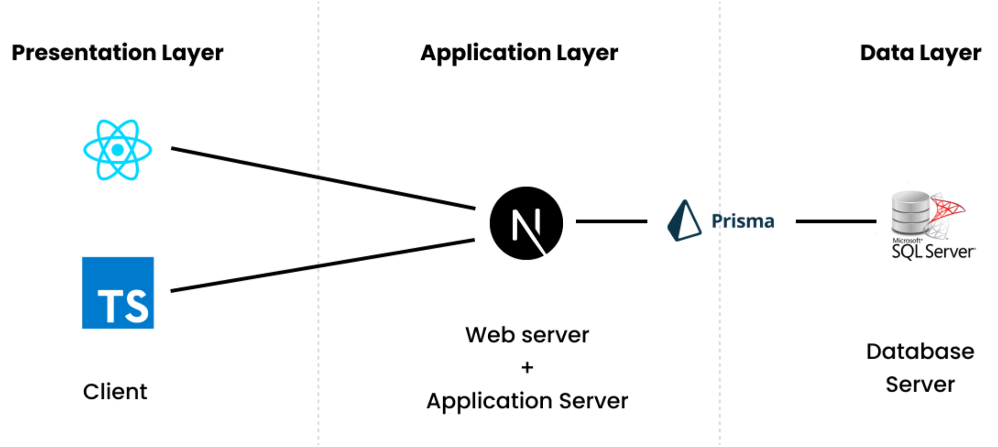


Figure 1: Layer diagram of the dispatch board system architecture

3.2.3 Difficulties encountered and their resolution

At the start of the project, we encountered difficulties with the initial setup. We initially considered creating a Docker container to deploy the project and the database in a local environment, ensuring consistency across team members. However, after conducting research and some trial and error, we determined that this would introduce unnecessary complexity.

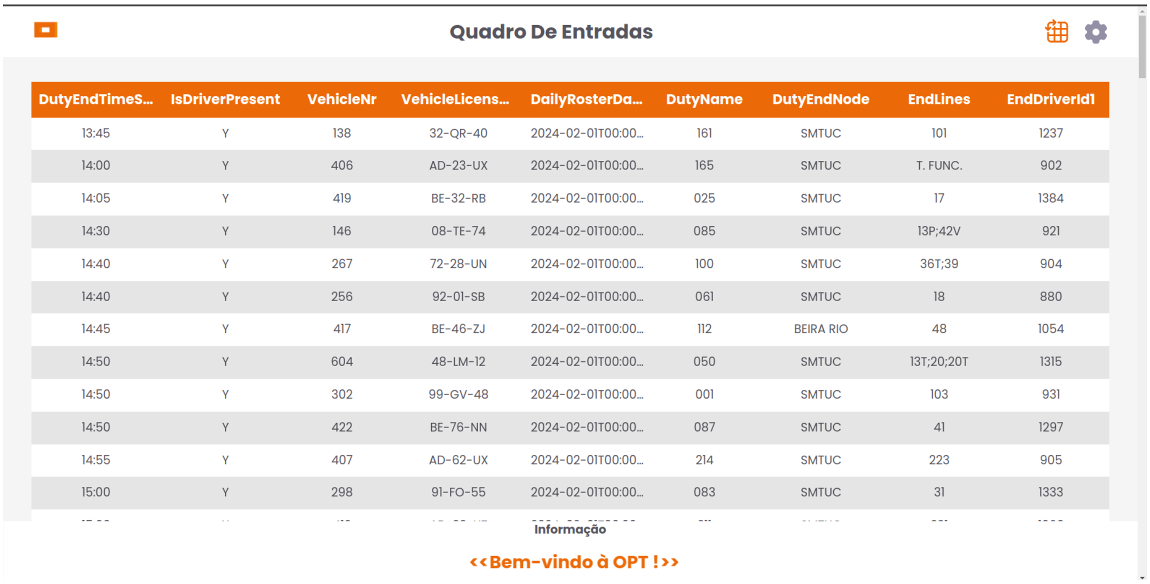
Fortunately, OPT provided us with a pre-existing database, eliminating the need to create one from scratch and simulate data. However, we faced the challenge of communicating with the database from both inside and outside of OPT. There were two different methods of connecting to the database, requiring manual changes to the .env file.

To address this issue, we developed a console executable file that automates the switching between environments, simplifying the process for seamless communication.

3.3 Developed Solution

3.3.1 Table of entries

We developed a comprehensive table of entries, encompassing all the necessary information for transport operators. Leveraging the existing table model used by OPT, we undertook a meticulous redesign to enhance its modernity, appeal, and user-friendliness. This revamped table features a header displaying the logo of the company utilizing the application, along with intuitive options for toggling between the table of entries and the table of exits. Additionally, users can easily navigate to the settings page, where they have access to a variety of customization options. These settings allow users to tailor the application's appearance and functionalities according to their preferences, including the ability to enable or disable specific features. The table dynamically displays all transport schedules from the current time until the end of the day, with real-time updates ensuring accuracy and reliability. Users are provided with clear visual cues, such as a red highlight for any delayed transport schedules, facilitating efficient and effective transport management. This system not only enhances operational efficiency but also significantly improves the user experience for all stakeholders involved. At the bottom of the page, there is an area dedicated to displaying the most recently updated information by the application user. This feature enables users to promptly report incidents such as road accidents, ensuring timely dissemination of critical information.



Quadro De Entradas

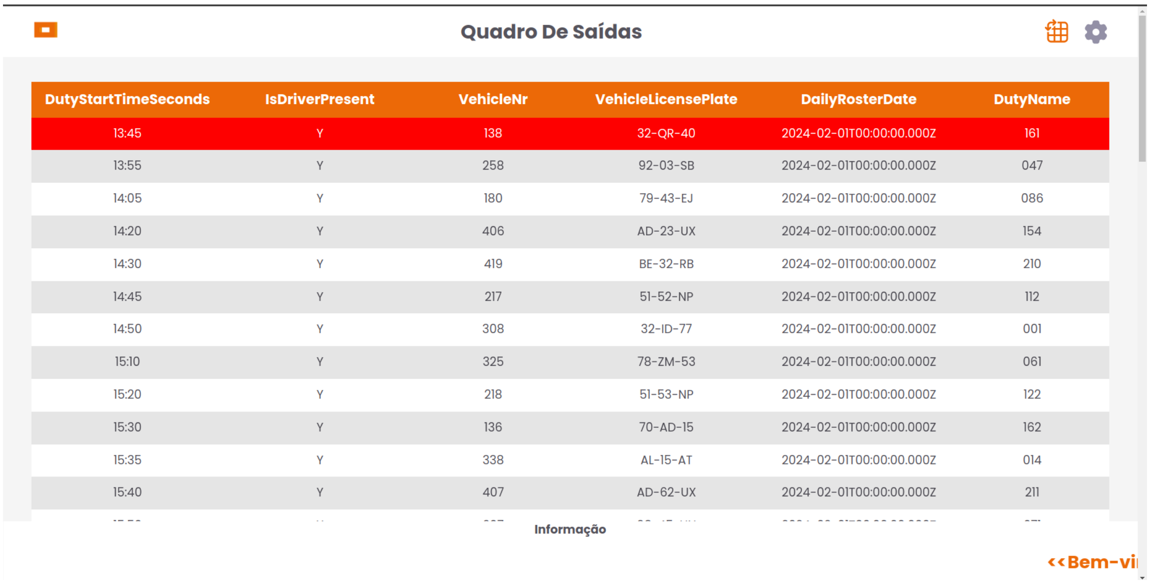
DutyEndTimeS...	IsDriverPresent	VehicleNr	VehicleLicens...	DailyRosterDa...	DutyName	DutyEndNode	EndLines	EndDriverId1
13:45	Y	138	32-QR-40	2024-02-01T00:00...	161	SMTUC	101	1237
14:00	Y	406	AD-23-UX	2024-02-01T00:00...	165	SMTUC	T. FUNC.	902
14:05	Y	419	BE-32-RB	2024-02-01T00:00...	025	SMTUC	17	1384
14:30	Y	146	08-TE-74	2024-02-01T00:00...	085	SMTUC	13P;42V	921
14:40	Y	267	72-28-UN	2024-02-01T00:00...	100	SMTUC	36T;39	904
14:40	Y	256	92-01-SB	2024-02-01T00:00...	061	SMTUC	18	880
14:45	Y	417	BE-46-ZJ	2024-02-01T00:00...	112	BEIRA RIO	48	1054
14:50	Y	604	48-LM-12	2024-02-01T00:00...	050	SMTUC	13T;20;20T	1315
14:50	Y	302	99-GV-48	2024-02-01T00:00...	001	SMTUC	103	931
14:50	Y	422	BE-76-NN	2024-02-01T00:00...	087	SMTUC	41	1297
14:55	Y	407	AD-62-UX	2024-02-01T00:00...	214	SMTUC	223	905
15:00	Y	298	91-FO-55	2024-02-01T00:00...	083	SMTUC	31	1333

Informação

<<Bem-vindo à OPT !>>

3.3.2 Table of exits

We have developed a table of exits with essential information tailored for transport operators. Modeled closely after the table of entries utilized by OPT, we have streamlined its layout while maintaining its modern and intuitive design. This revised table includes a header featuring the company logo of the application user, offering seamless navigation between the table of exits and the table of entries. Similar to its counterpart, users can access the settings page to customize the application's appearance and functionality according to their preferences. The table of exits presents updated transport schedules from the current time until the end of the day, ensuring real-time accuracy. Visual indicators promptly highlight any delays or disruptions in service, facilitating efficient transport management and enhancing user experience. At the bottom of the page, users also have an area to provide updates, such as reporting road incidents, ensuring comprehensive and timely information dissemination.

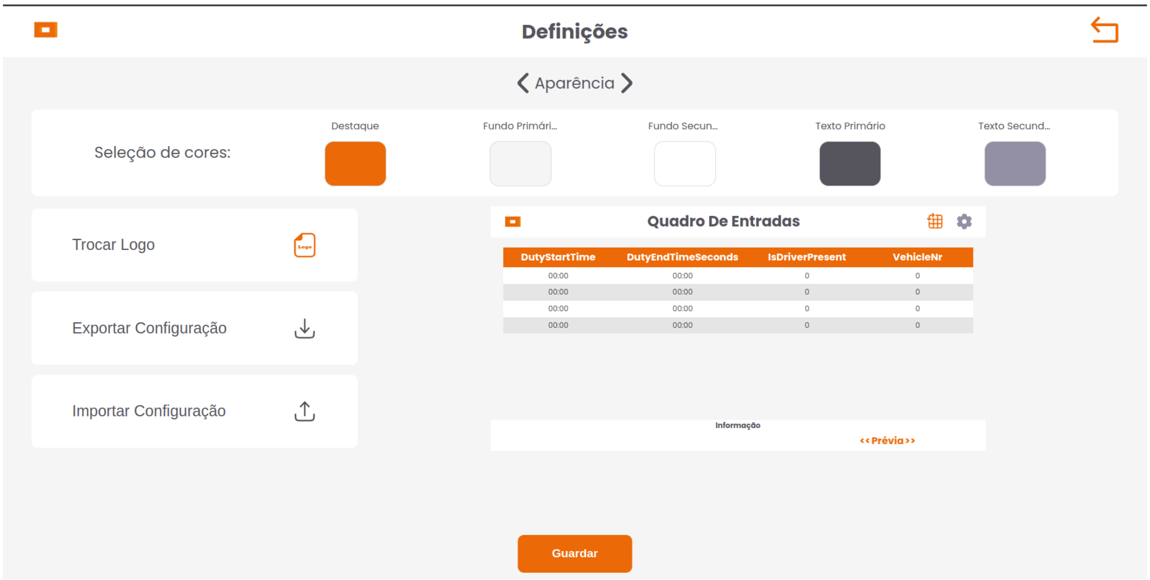


Quadro De Saídas

DutyStartTimeSeconds	IsDriverPresent	VehicleNr	VehicleLicensePlate	DailyRosterDate	DutyName
13:45	Y	138	32-QR-40	2024-02-01T00:00:00.000Z	161
13:55	Y	258	92-03-SB	2024-02-01T00:00:00.000Z	047
14:05	Y	180	79-43-EJ	2024-02-01T00:00:00.000Z	086
14:20	Y	406	AD-23-UX	2024-02-01T00:00:00.000Z	154
14:30	Y	419	BE-32-RB	2024-02-01T00:00:00.000Z	210
14:45	Y	217	51-52-NP	2024-02-01T00:00:00.000Z	112
14:50	Y	308	32-ID-77	2024-02-01T00:00:00.000Z	001
15:10	Y	325	78-ZM-53	2024-02-01T00:00:00.000Z	061
15:20	Y	218	51-53-NP	2024-02-01T00:00:00.000Z	122
15:30	Y	136	70-AD-15	2024-02-01T00:00:00.000Z	162
15:35	Y	338	AL-15-AT	2024-02-01T00:00:00.000Z	014
15:40	Y	407	AD-62-UX	2024-02-01T00:00:00.000Z	211

3.3.3 Settings page

Our settings page offers extensive customization options tailored to enhance user interaction. A sleek color selection bar allows precise configuration of highlighting, primary and secondary backgrounds, as well as primary and secondary text colors. Each option triggers a color picker for seamless color selection and adjustment. Users can upload a company logo effortlessly using a dedicated upload button. For configuration management, the page includes export/import functionality for saving and reapplying settings as a JSON file. A preview section provides a live demonstration of layout changes, ensuring users can visualize and confirm adjustments to entry and exit tables before finalizing. This comprehensive approach ensures a personalized and intuitive experience, empowering users to tailor the application's appearance and functionality to their exact specifications.



Seleção de cores:

Destaque	Fundo Primári...	Fundo Secun...	Texto Primário	Texto Secund...
🟧	⬜	⬜	⬛	⬛

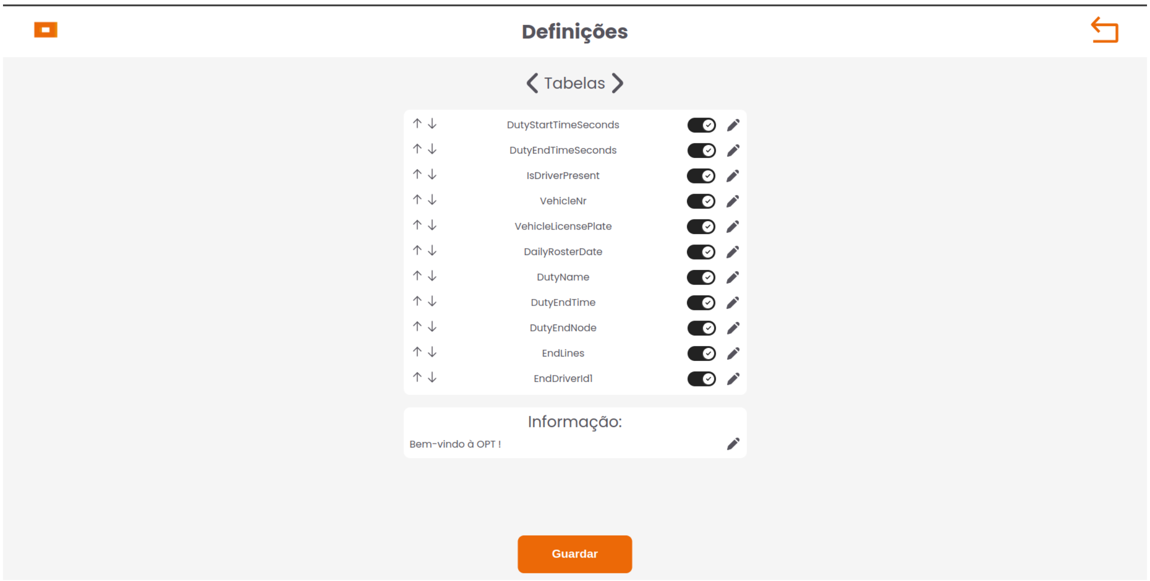
Quadro De Entradas

DutyStartTime	DutyEndTimeSeconds	IsDriverPresent	VehicleNr
00:00	00:00	0	0
00:00	00:00	0	0
00:00	00:00	0	0
00:00	00:00	0	0

Informação

<<Prévia>>

The second settings page focuses on fine-tuning table display and content preferences. Users have granular control over column visibility, allowing them to show or hide specific columns as needed. Additionally, columns can be renamed to align with specific data or organizational needs. Display order customization empowers users to arrange columns in a preferred sequence, optimizing data presentation. Furthermore, users can modify the information displayed at the bottom of entry and exit tables, ensuring relevance and clarity. This page serves as a powerful tool for configuring and optimizing data presentation, enhancing usability and efficiency in managing transport operations. By providing flexible customization options, it supports diverse user requirements and preferences.



↑ ↓	DutyStartTimeSeconds	✓	✎
↑ ↓	DutyEndTimeSeconds	✓	✎
↑ ↓	IsDriverPresent	✓	✎
↑ ↓	VehicleNr	✓	✎
↑ ↓	VehicleLicensePlate	✓	✎
↑ ↓	DailyRosterDate	✓	✎
↑ ↓	DutyName	✓	✎
↑ ↓	DutyEndTime	✓	✎
↑ ↓	DutyEndNode	✓	✎
↑ ↓	EndLines	✓	✎
↑ ↓	EndDriverId1	✓	✎

Informação:

Bem-vindo à OPT !

3.4 Validation

To ensure our project aligned with the requirements set by OPT, we sought guidance from our project manager, who provided weekly feedback for improvements. Our application was designed for larger screens, so testing it on the TVs at OPT headquarters was crucial. This allowed us to better gauge the dimensions of each compartment and validate our approach.

4 Conclusões

4.1 Resultados alcançados

Sumariar os resultados alcançados e contribuições (em relação aos objetivos).

No caso de trabalho em grupo, clarificar as contribuições individuais, em termos qualitativos e quantitativos (percentagem).

4.2 Lições aprendidas

Refletir sobre as lições aprendidas (tendo em conta os objetivos de aprendizagem).

4.3 Trabalho futuro

Ideias de melhorias e trabalho futuro.