Darkstore API

Software Requirements Specification

Contents

Introduction

Purpose

The purpose of this project is to create a REST API service that manages couriers, orders, and their distribution for a darkstore. The service aims to optimize the courier workflow, manage orders, and streamline the allocation process, ensuring smooth operations and timely deliveries within the darkstore ecosystem.

System restrictions

- The system will not provide real-time tracking of courier locations.
- The system will not support chat or notification features.
- The service will not support multiple languages; all data and communication will be in the same language.

Intended audience and project scope

1. **Administrators / Logistics Managers:**
    - *Function:* Responsible for managing and organizing courier operations, such as assigning orders to couriers, monitoring delivery progress, and maintaining courier schedules.
    - *Access:* Full system access, including order assignment, tracking, and data management.
2. **Couriers:**
    - *Function:* Responsible for executing the deliveries, viewing assigned orders, updating order status upon delivery, and managing their availability schedules.
    - *Access:* Restricted access limited to the orders assigned and their delivery status updates.

Potential improvements

- Real-time order tracking
- Machine learning for demand prediction
- Registration of new couriers via app

Functional Requirements

- **FR1 - REST API Implementation** - The system must implement a REST API service, including endpoints for courier registration, order addition, and distribution.
- **FR2 - Courier Rating Calculation** - The service must include functionality to calculate courier ratings as detailed in the project description.
- **FR3 - Rate Limiter Implementation** - The system should incorporate a rate limiter to restrict API requests to 10 RPS per endpoint.
- **FR4 - Order Distribution Algorithm** - The service must implement an algorithm for distributing orders among couriers to minimize delivery costs.
- **FR5 - Courier Registration** - The system must allow uploading a list of couriers with their work schedules and designated areas via a JSON format.
- **FR6 - Retrieve Courier Information** - The system should provide information about individual couriers and all couriers, with pagination support.
- **FR7 - Retrieve Order Information** - The system should provide details about individual orders and all orders, with pagination support.
- **FR8 - Order Assignment** - The system must include a method to assign orders to couriers at the start of their workday.

Non-Functional Requirements

- Performance requirements
    - The application will use a stable database system that can handle higher current load
    - The application will take into account the growth of users and the growing volume of stored data.
- Security requirements
    - Login credentials will be encrypted and secured
    - The system will use an external payment gateway to transfer funds
- Quality requirements
    - The design of the application will be intuitive, modern and minimalistic
    - The application will be usable without limitations on all common operating systems and browsers

References

https://gitlab.fel.cvut.cz/B231_B6B36EAR/morenmat

Class diagram

Only the highlighted (blue) lines in the class diagram will be implemented.