Курсов проект

На

**Димитър Кючуков, фак. № 471222046**

**Гергана Роева, фак. № 471222013**

Тема: „Система за проследяване и управление на куриерски пратки“

Функционална Спецификация

Общо представяне

Системата за проследяване и управление на куриерски пратки е предназначена да автоматизира и оптимизира процесите на куриерските услуги, като предоставя точна и актуална информация за статуса на пратките в реално време. Тя трябва да поддържа функционалности както за куриерите, така и за клиентите, включително управление на пратки, проследяване на статуса, идентификация и управление на потребители, и интеграция с външни системи за плащания, адресна информация и други.

Основни функционалности

За клиентите:

Регистрация и вход: Потребителите се регистрират в системата, като предоставят необходимата информация. След регистрация могат да влизат в своя акаунт чрез потребителско име и парола.

Създаване на пратка: Клиентите могат да създават нова пратка, като посочват отправна и крайна точка, размери, тегло и допълнителни изисквания (напр. застраховка).

Проследяване на пратка: Възможност за проследяване на статуса на пратката в реално време, включително местоположение и очаквана дата и час за доставка.

История на пратките: Потребителите могат да преглеждат историята на своите пратки, включително успешно доставени и текущо обработвани.

За куриерите:

Управление на задачите за доставка: Куриерите виждат списък с пратки, които трябва да бъдат доставени, със съответната информация за всяка пратка и маршрут за оптимална доставка.

Обновяване статуса на пратка: Възможност за обновяване на статуса на пратка (напр. приета, в процес на доставка, доставена).

Сканиране на пратки: Интеграция с баркод скенери за бързо идентифициране и обработка на пратки.

За администраторите:

Управление на потребители: Добавяне, премахване и редактиране на потребителски акаунти.

Отчети и анализи: Генериране на отчети за активността на системата, включително брой доставки, време за доставка, и други метрики за производителност.

Конфигурация на системата: Настройка на параметрите на системата, включително тарифи за доставка, настройки за сигурност и интеграции с външни услуги.

Интеграции:

Платежни системи: Интеграция с платежни системи за обработка на плащания за доставки.

Системи за адресна информация: Интеграция със системи за адресна информация за автоматизиране на въвеждането на адреси и оптимизация на маршрутите за доставка.

Външни API-та за логистика: Взаимодействие с външни логистични и куриерски услуги за разширяване на обхвата на доставки и услуги.

Заключение

Тази функционална спецификация очертава основните компоненти и функции, които трябва да бъдат разработени за системата за проследяване и управление на куриерски пратки. Следващата стъпка е да се разработи техническа спецификация, която да детайлизира технологичните решения, необходими за реализацията на тези функции.

Техническа Спецификация

За разработването на система за проследяване и управление на куриерски пратки, техническата спецификация обхваща избора на технологии, архитектурата на системата и важни технически решения. В този контекст, ето пример за техническа спецификация:

Архитектурен обзор

Системата ще използва микросервизна архитектура, която позволява модуларност, лесна мащабируемост и независимо развитие и разгръщане на отделните компоненти. Микросервизите ще комуникират помежду си чрез леки API заявки, обикновено REST или GraphQL за уеб интерфейси и gRPC за вътрешни сервизни взаимодействия.

Технологичен стек

Сървърна страна: Node.js / Express за бърза разработка и лесна интеграция с различни бази данни и външни услуги. Алтернативно, за по-високо ниво на производителност и типизация, може да се използва Go или Rust.

Клиентска страна: React.js за разработка на динамични и отзивчиви уеб интерфейси. Vue.js или Angular са валидни алтернативи в зависимост от предпочитанията на екипа.

База данни: PostgreSQL за основна транзакционна база данни поради нейната надеждност, производителност и поддръжка на комплексни заявки. MongoDB може да се използва за документно-ориентирани данни или кеширане.

Съобщения и асинхронна комуникация: RabbitMQ или Apache Kafka за управление на асинхронни съобщения и процеси, осигуряващи висока достъпност и разпределена обработка.

Кеширане: Redis за кеширане на често достъпвани данни и сесии, което намалява натоварването върху основната база данни и ускорява времето за отговор.

Автентикация и сигурност: OAuth 2.0 и JWT за управление на достъпа и аутентикация на потребителите. Използване на HTTPS за криптиране на трафика и Hashing (напр. bcrypt) за съхранение на пароли.

Инфраструктура и разгръщане: Docker за контейнеризация на приложения и Kubernetes за оркестрация, осигуряващи лесно разгръщане и мащабируемост в облака (AWS, Google Cloud или Azure).

Фронтенд и бекенд интеграция: GraphQL за оптимизиране на заявките между клиент и сървър, намалявайки излишните данни и подобрявайки производителността на приложението.

Основни компоненти и техните отговорности

Проследяване на пратки: Микросервиз за управление на статуси и локации на пратки в реално време.

Управление на пратки: Сервиз за създаване, редактиране и анулиране на пратки, включително управление на маршрути и доставки.

Потребителски интерфейс: Клиентски приложения за уеб и мобилни устройства, предоставящи интерфейси за клиенти и куриери.

Управление на потребители: Сервиз за аутентикация, регистрация и управление на потребителски профили.

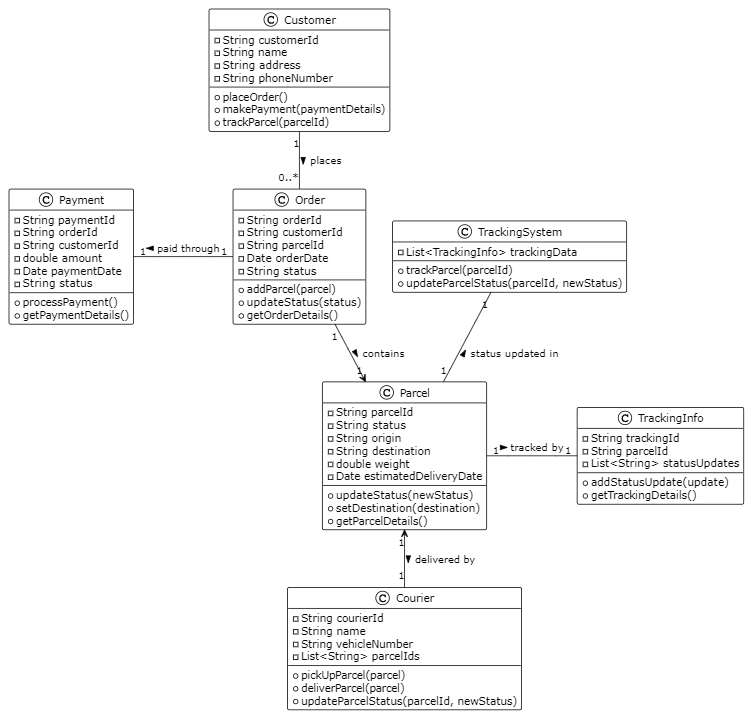
Отчети и анализи: Сервиз за генериране на отчети и аналитични данни за дейността на системата.

Заключение

Изборът на технологии и архитектура е съобразен с нуждите за мащабируемост, надеждност и лесно управление на системата за проследяване и управление на куриерски пратки. Микросервизният подход позволява гъвкаво развитие и поддръжка на системата, докато избраният технологичен стек осигурява баланс между производителност, разработка и лесна интеграция.

Архитектурни диаграми

1. Class diagram



1. State diagram

A diagram of a delivery process

Description automatically generated

1. Package diagram

A diagram of a parcel management

Description automatically generated

1. Component diagram

A diagram of a software

Description automatically generated

1. Use case diagram

A diagram of a company's flowchart

Description automatically generated

Линк към сорс кода за диаграмите: <https://github.com/mitakawe123/SA_Diagrams>