

НАЦИОНАЛЬНО-ИССЛЕДОВАТЕЛЬСКАЯ КОРПОРАЦИЯ ИТМО



Факультет программной инженерии и компьютерной техники

Информационные системы и базы данных

Курсовая работа

Выполнили
студенты:

Стуков Егор Александрович

Нестеров Иван Алексеевич

Группа № Р33302

Преподаватель:

Гаврилов Антон Валерьевич

г. Санкт-Петербург

2022

Первый этап:

Источник для описания предметной области:

<https://deathstranding.fandom.com/wiki/Orders>



Описание предметной области на основе источника:

Система доставки грузов организации «Bridges» принимает от пользователей заявки на доставку груза из одного города-бункера в другой. Полученные заявки содержат информацию о весе груза, габаритах транспортного контейнера, пункте назначения, пункте выдачи заказа, его объявленной стоимости, а также дополнительные пометки, характеризующие особенности доставки («хрупкий груз», «не погружать в воду» и т. п.).

На основе полученных данных формируется запрос груза на доставку, доступный всем курьерам для обработки. Помимо введенной пользователем информации, запрос содержит информации об объеме награды за выполнение доставки и статусе заказа («Новый», «Доставляется», «Доставлен», «Утерян», «Уничтожен»).

Любой курьер может взять заказ в обработку. При этом статус заказа меняется на «Доставляется». У курьера одновременно может быть в обработке несколько заказов. При успешной доставке груза статус заказа меняется на «Доставлен», курьер получает на свой счёт награду за выполнение доставки, размер награды зависит от нескольких факторов: дистанция, время выполнения, состояние груза.

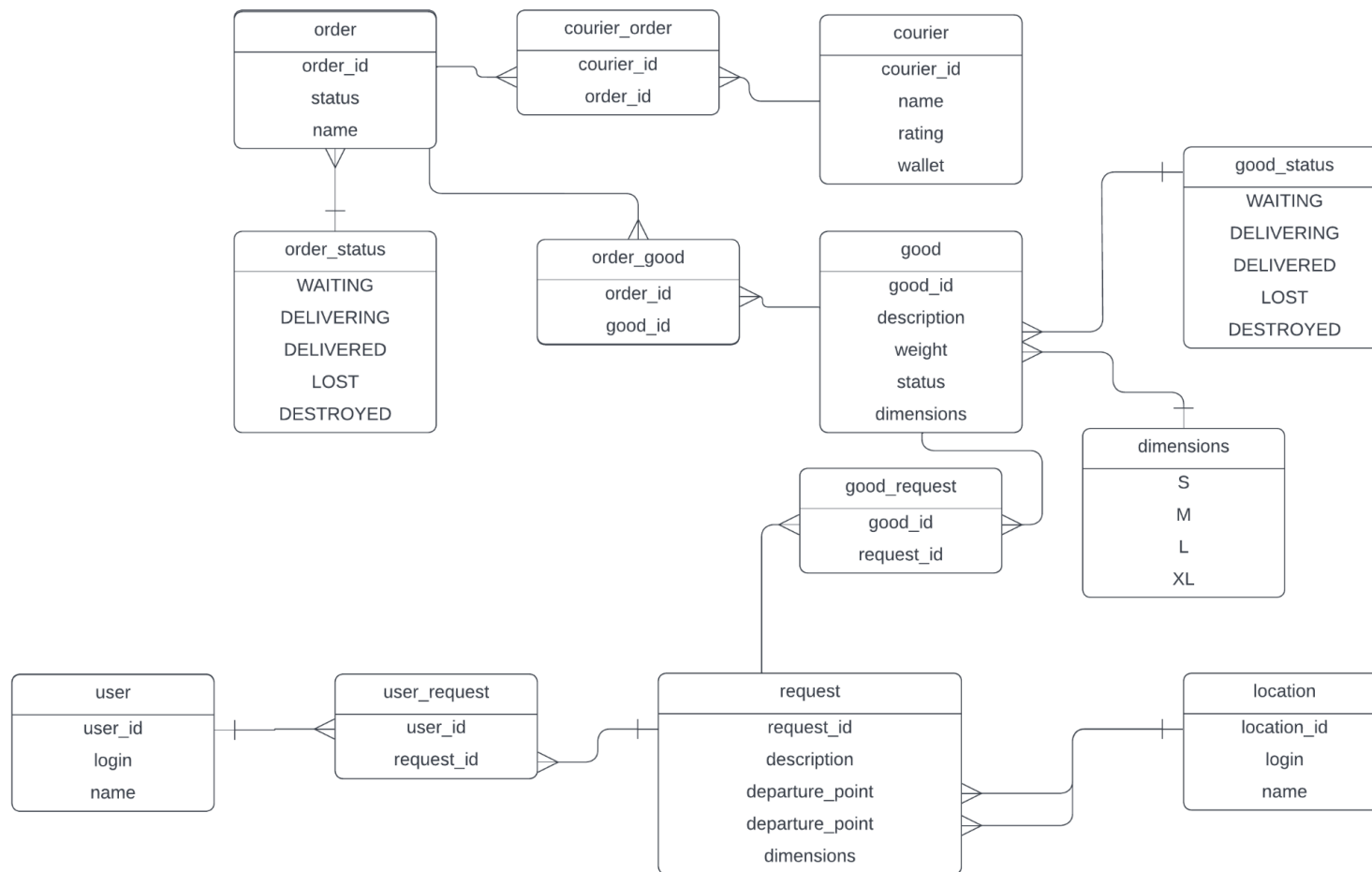
В случае нападения на курьера, природного катаклизма или под влиянием иных отрицательных факторов заказ может быть утерян или уничтожен. Заказ получает соответствующий статус. В случае установления статуса «Утерян», в карточке заказа указывается ближайший к месту утери груза пункт. В случае установления статуса «Уничтожен» клиенту выплачивается компенсация в размере указанной ценности груза.

Если курьер потерял заказ, система назначает ему штраф, размер которого зависит от стоимости груза и рейтинга курьера.

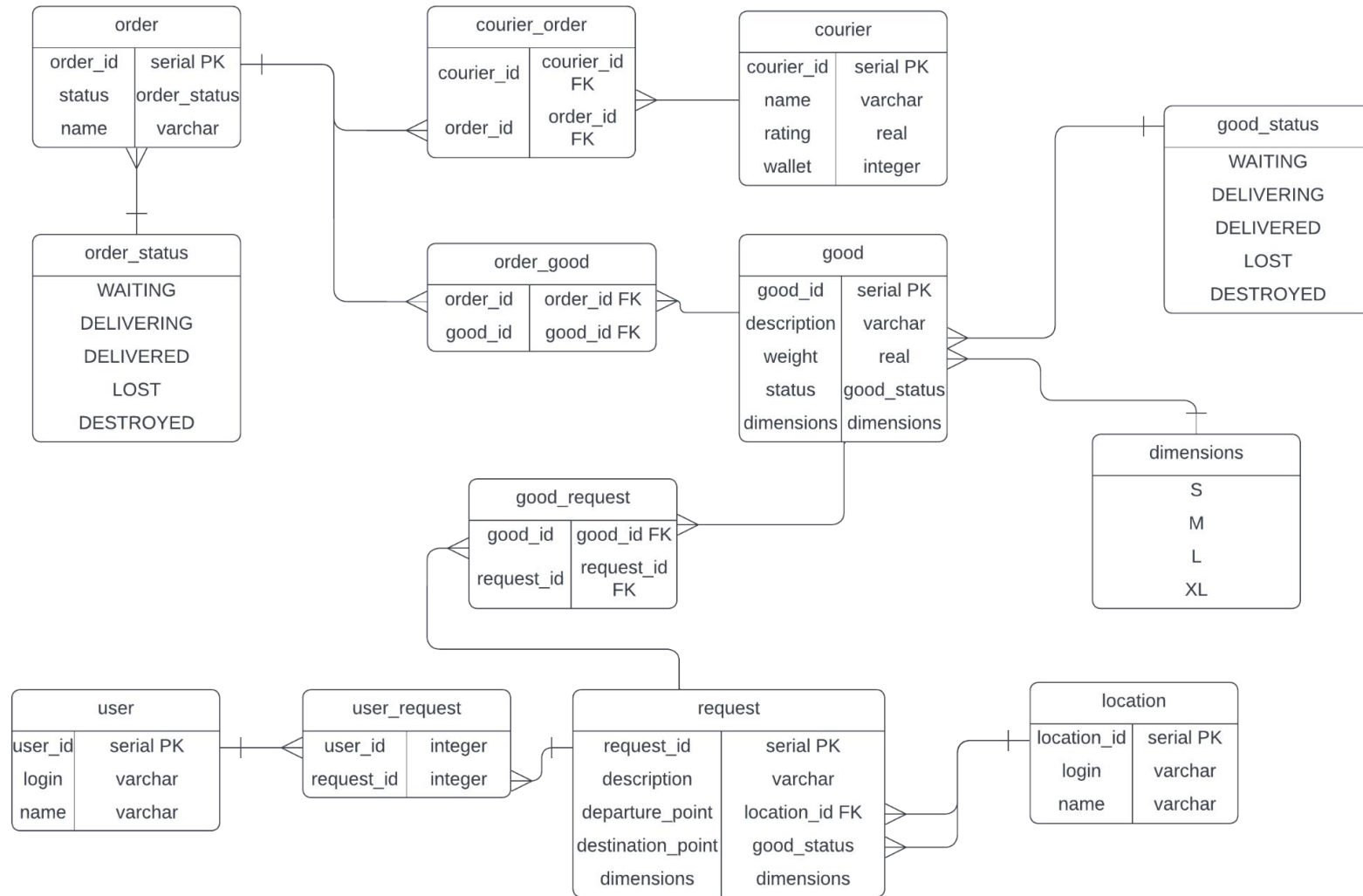
Курьеры могут возвращать утерянные грузы. В таком случае заказ получает статус «Доставлен», а курьер, доставивший груз, получает награду за доставку.

Второй этап:

ER-диаграмма предметной области:



Даталогическая модель на основе ER-модели:



Третий этап:

В ходе третьего этапа был выполнен некоторый набор действий: опишем каждое из них подробнее.Итак:

1. Добавлены комплексные ограничения целостности:
 - a. В одном заказе не может находиться более 10 товаров
 - b. Курьеры с рейтингом ниже 3.5 не могут доставлять товары с дополнительными требованиями (хрупкий, не трясти и т.д.)
2. Придуманы и реализованы следующие процедуры:
 - a. Автоматическое опреописаделение размеров груза (XL, L, S...) на основе указанных габаритов.
 - b. Автоматическое изменение рейтинга курьера на основе изменения статуса товара.
3. Разработаны триггеры, вызывающие указанные выше процедуры при инициации изменений в отслеживаемых таблицах
4. Выявлены наиболее частые операции к нашей схеме данных (на основе бизнес-логики):
 - a. Вывод курьерами заказов по статусу, весу, локации
 - b. Поиск клиентом курьера по номеру заказа
 - c. Поиск клиентом курьеров по рейтингу
5. На основе анализа наиболее часто использующихся операций разработаны индексы, для оптимизации соответствующих запросов
6. Описанные в предыдущих этапах словесные ограничения переписаны в соответствующий скрипт на создание нужных сущностей с необходимыми констрантами. В скриптах исползуются подзапросы для динамической связи сущностей между собой
7. Написан скрипт на заполнение таблицы тестовыми данными, в добавлении данных применяются особенности описания сущностей из пункта выше (внесение ID во вторую таблицу на основе его поиска в предыдущей таблице по указанным параметрам)
8. Написан скрипт на очистку данных всех таблиц, учитывающий возможность только каскадного удаления некоторых данных
9. Написан скрипт на удаление всех используемых сущностей из схемы
10. В связи с обилием появившегося кода и неудобством его чтения внутри документа, [был создан GitHub репозиторий](#), в котором отражена реализация всех описанных выше пунктов на языке SQL Репозиторий будет динамически обновляться по продолжении работы над курсовой работой.

Четвертый этап:

В ходе четвертого этапа работы было написано веб-приложение с бекенд-частью, основанной на фреймворке Spring Boot и фронтенд-частью, основанной на использовании библиотеки React. Была автоматизирована вся происходившая ранее в консоли работа, а так же добавлен веб-интерфейс. Пользователь имеет возможность авторизоваться как курьер или как заказчик. Далее каждый из них получает доступ к своей домашней странице и возможностям: для пользователя – создать новый заказ, отслеживать статус своего заказа, а курьер – получить список доступных для взятия заказов (как новых, так и, например, утерянных другим курьером), отслеживать свои активные заказы, менять статус заказа, чтобы уведомить заказчика о том, что заказ доставлен, утерян или уничтожен. Фронтенд-часть отправляет бекенд-части нужные данные в виде JSON DTO, где они обрабатываются. Решается также проблема взаимосвязи таблиц между собой. Инструменты Spring JDBC осуществляют все задачи бизнес-логики, которые ранее выполнялись в консоли нами. Обратно на фронтенд-часть в ряде случаев возвращаются данные, которые на экране нужно обновить. Весь исходный код выложен в указанный выше GitHub-репозиторий, где так же можно отследить историю коммитов, что, на наш взгляд, является более исчерпывающим описанием хода работы.