

**Министерство науки и высшего образования Российской
Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)**

Факультет Инфокоммуникационных технологий (ИКТ)

Образовательная программа Мобильные и сетевые технологии

Дисциплина: Бек-энд разработка

ОТЧЕТ

Лабораторная работа 4

Выполнил:

Рейнгеверц В.А.

Группа:

К33401

Проверил:

Добряков Д. И.

Санкт-Петербург
2023

Laboratory Work 4

| К33401 - Рейнгеверц В.А.

Requirements

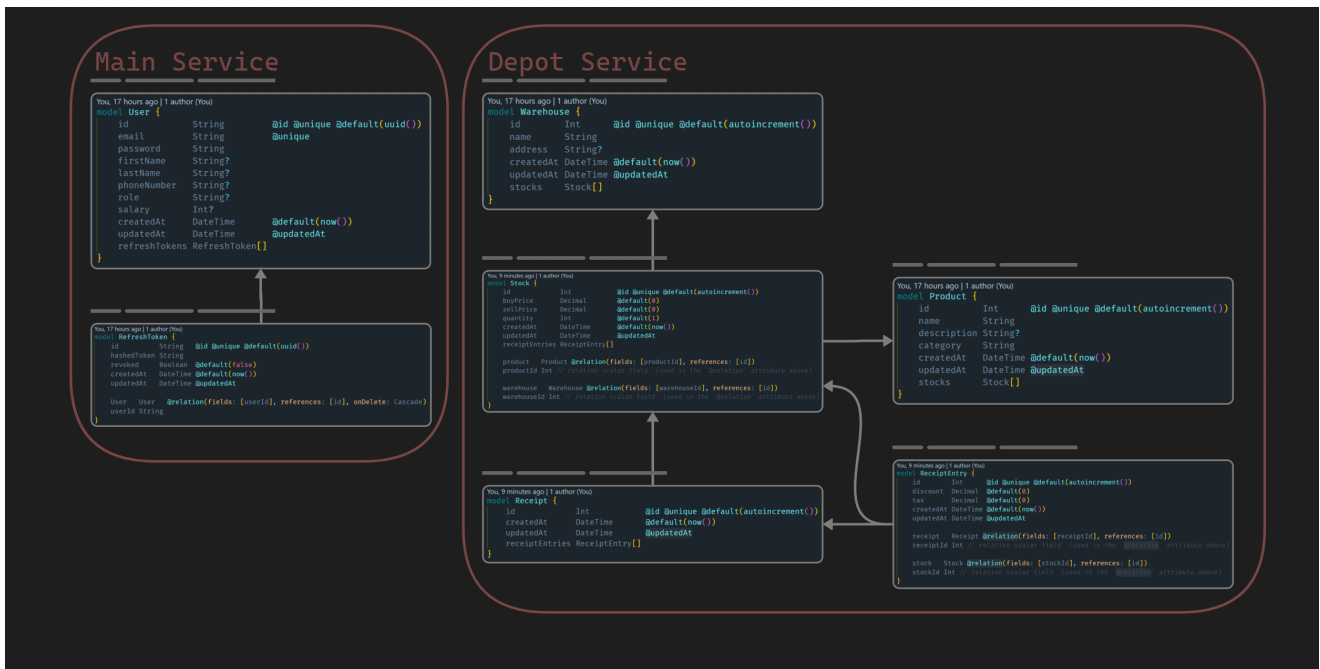
Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения. Делать это можно как с помощью docker-compose так и с помощью docker swarm. При разумном использовании swirl вы получите дополнительные баллы.

Description

| Вариант 4 - Сайт администратора интернет-магазина

- Вход
- Регистрация
- Учёт товара на складе
- Графики по продажам тех или иных товаров, по общей выручке предприятия
- Управление сотрудниками
- Реализованы 2 микросервиса, выполняющий содержательные функции

ER Diagram



Running in Docker

Delete all containers

```
docker rm $(docker ps -a -q)
```

Without `docker-compose`

Creating a network

```
docker network create lab4
```

```
docker network ls
```

```
docker network inspect lab4
```

Main Microservice

[Dockerfile](#)

Dockerfile -> Image

```
docker build --tag liprikon/lab4-main-service:1.0 ./mainService
```

Image -> Container

```
docker run -t -i -p 3333:3010 \  
  --rm --name main \  
  --network lab4 \  
  --volume dbMain:/container/src/db/data \  
  liprikon/lab4-main-service:1.0
```

- `-p` forwards container's 3010 port to 3333 port of host machine
- `-ti` allows CTRL + C to stop container
- `--rm` removes container after exit
- `--name` becomes hostname for containers
- `--network` adds container to network
- `--volume` makes database data persistent by [creating a named volume](#)

Depot Microservice

| [Dockerfile](#)

Dockerfile -> Image

```
docker build --tag liprikon/lab4-depot-service:1.0 ./depotService
```

Image -> Container

```
docker run -t -i \  
  --rm --name depot \  
  --network lab4 \  
  --volume dbDepot:/container/src/db/data \  
  liprikon/lab4-depot-service:1.0
```

- `-ti` allows CTRL + C to stop container
- `--rm` removes container after exit
- `--name` becomes hostname for containers
- `--network` adds container to network
- `--volume` makes database data persistent by [creating a named volume](#)

With docker-compose

| [docker-compose.yml](#)

```
docker-compose build && docker-compose up
```

- Main Microservice available at <http://localhost:3333>

Running without Docker

```
sh run.sh install
```

```
sh run.sh migrate
```

```
sh run.sh
```

- Main Microservice available at <http://localhost:3010>
- Depot Microservice available at <http://localhost:3020>

Port and host can be specified in [run.sh](#):

```
# <...>

# Main microservice
export MAIN_HOST="127.0.0.1"
export MAIN_PORT=3010

# Depot microservice
export DEPOT_HOST="127.0.0.1"
export DEPOT_PORT=3020

# <...>
```

Structure

| Repository design pattern

Main Microservice

► Unfold to see the list of all API Routes of Main Microservice

- Handles registration, login in, password recovery of a user
- Proxies requests to depotService

Depot Microservice

- ▶ Unfold to see the list of all API Routes of Depot Microservice
 - Provides warehouse, product, stock information

Gateway

- Main Service proxies authenticated `/depot/*` requests to Depot Service's `/*`
- Gateway is at [mainService/src/core/app.ts](#)

Microservice structure:

- Microservice entry point at [<ServiceName>/src/core/app.ts](#)
- Models are defined at [<ServiceName>/src/db/schema.prisma](#)
- Controllers are defined at [<ServiceName>/src/controllers/*](#)
- Middlewares are defined at [<ServiceName>/src/middleware/*](#)
- Routes are defined at [<ServiceName>/src/routes/*](#)
- Services are defined at [<ServiceName>/src/services/*](#)
- Utility functions are defined at [<ServiceName>/src/utils/*](#)