

**МИНИСТЕРСТВА НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
Санкт-Петербургский политехнический университет Петра Великого

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК
ВЫСШАЯ ШКОЛА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Курсовой проект

Наименование: Sweets Shop с использованием «Spring Framework»
по курсу: "«Программное обеспечение распределённых
вычислительных систем»"

Выполнил студент гр. 3540901/91501

Селиверстов С.А.

Руководитель

Стручков И.В.

«__» _____ 2020г.

Санкт-Петербург
2020

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ.....	2
Глава 1. Анализ задания	3
1. Формулировка индивидуального задания	3
2. Требования, которым должно удовлетворять приложение	3
3. Основные варианты использования.....	3
4. Описание модели предметной области	4
5. Вовлеченные Сущности.....	4
Глава 2. Реализация задания с помощью «Spring Framework»	5
1. Объектно-ориентированное проектирование с учётом особенностей технологии	5
2. Диаграмма последовательности	6
3. Тестирование	7
4. Интерфейс приложения	7
5. Инструкция системному администратору по развёртыванию приложения	9
6. Ведение базы данных заказов	10
7. Инструкция пользователю по запуску приложения	11
Вывод.....	12
Литература.....	13

Глава 1. Анализ задания

1. Формулировка индивидуального задания

Реализовать приложение «Sweets Shop» на языке Kotlin с помощью Spring Framework. Система Sweets Shop предназначена для автоматизации процесса кондитерских услуг на заказ. Данная система предоставляет владельцу сайта найти клиентов, желающих купить кондитерские изделия и повара, способного выполнить его.

2. Требования, которым должно удовлетворять приложение

Приложение должно строиться, используя следующие принципы и инструменты:

- Spring Framework
- Rest
- Паттерны

3. Основные варианты использования

1. Создание заказа

1. Клиент открывает главную страницу, на которой генерируется список заказов
2. Клиент генерирует заказ
3. Клиент оформляет заказ
4. Клиент подтверждает оформление заказа.

Примечание. Для простоты демонстрации реализован генератор заказов

2. Просмотр заказа оператором

1. Оператор выполняет вход в систему
2. Оператор просматривает список заказов, которые ожидают его сопровождения

3. Оператор выбирает заказ для сопровождения
4. Оператор звонит клиенту для точного подтверждения
5. Оператор выставляет статус заказа.

3. Выполнение заказа

1. Повар выполняет вход в систему
2. Повар просматривает список заказов, которые готовы и ожидают выполнения
3. Повар выбирает заказы, которые хочет выполнить
4. Повар смотрит информацию
5. Повар принимает оплату от клиента.

4. Описание модели предметной области

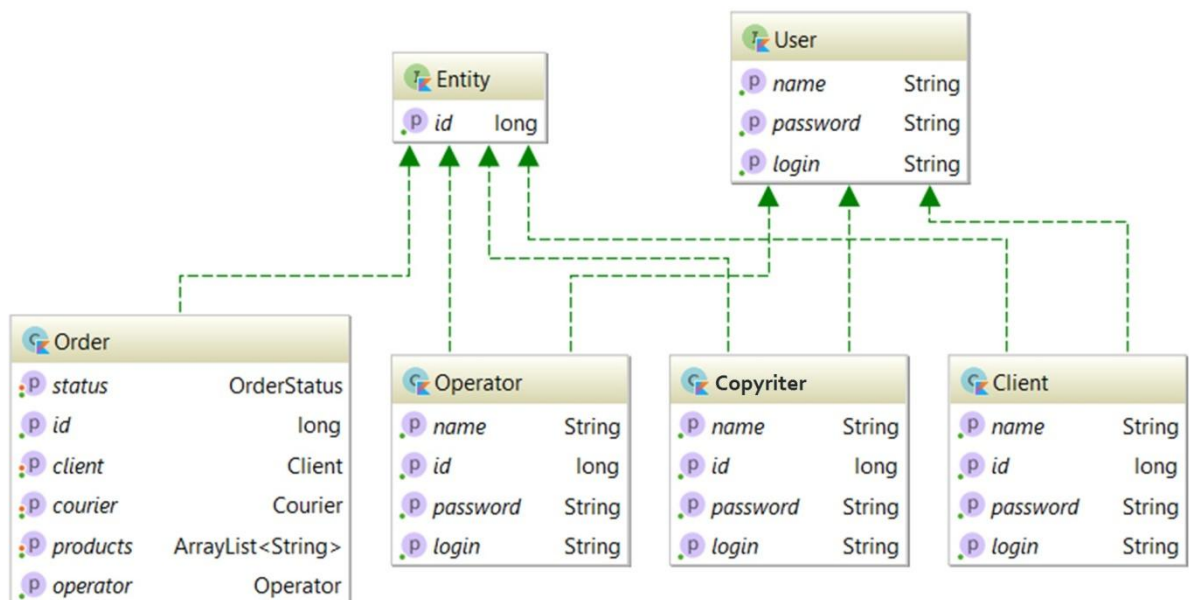


Рис.1

5. Вовлеченные Сущности

В результате разработки вариантов использования были выделены следующие вовлеченные сущности: Клиент, Оператор, Повар, Заказ.

Глава 2. Реализация задания с помощью «Spring Framework»

1. Объектно-ориентированное проектирование с учётом особенностей технологии

Диаграммы классов представлены на рис.2

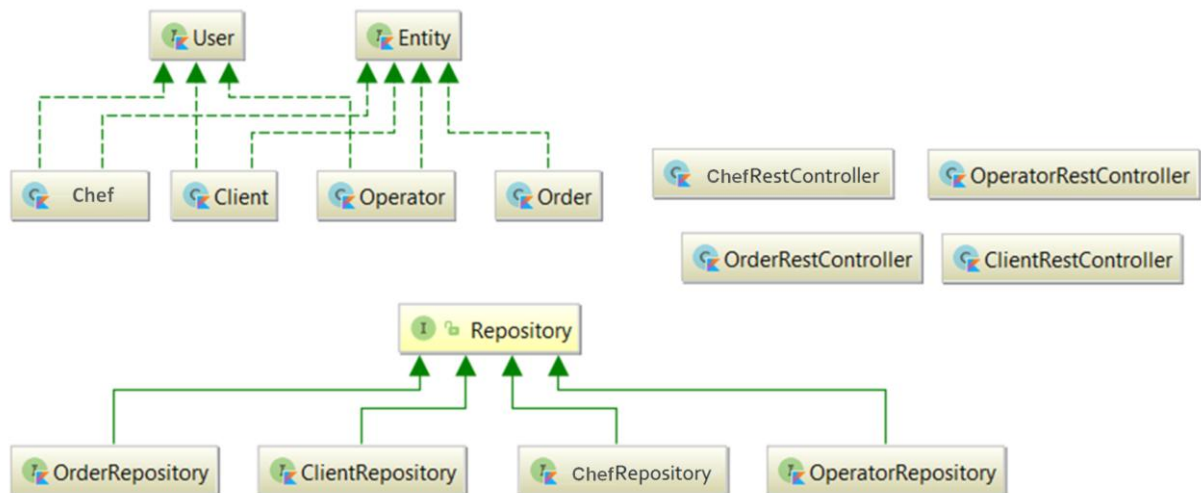


Рис.2

@SpringBootApplication – сканирует все контроллеры, сервисы и прочие составляющие спринга, чтобы все это вместе запустить

Application – класс запуска программы

@RestController – принимает запросы пользователя и возвращает данные в JSON формате

OrderRestController – контроллер для работы с сущностью заказа

@Entity – это аннотация, который берет соответствующий класс и на основе его полей делает таблицу в БД.

Order – класс заказа

User – класс пользователя

CrudRepository – обеспечивает сложные функциональные возможности CRUD для класса сущностей

2. Диаграмма последовательности

Диаграммы последовательности представлены на рис.3-5.



Рис.3

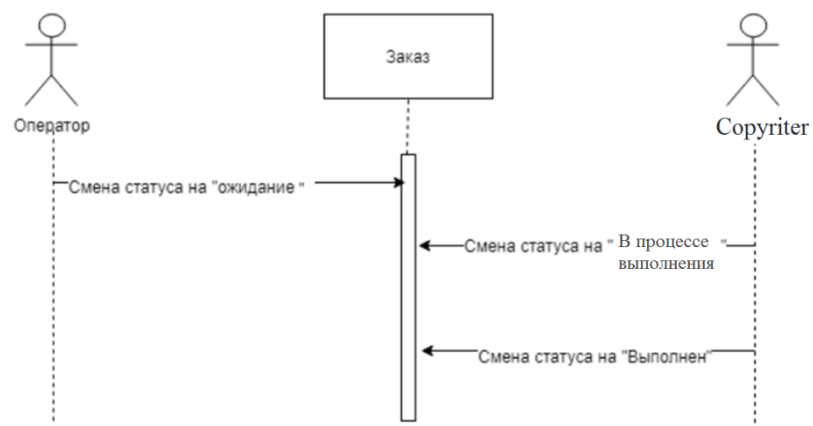


Рис.4



Рис.5

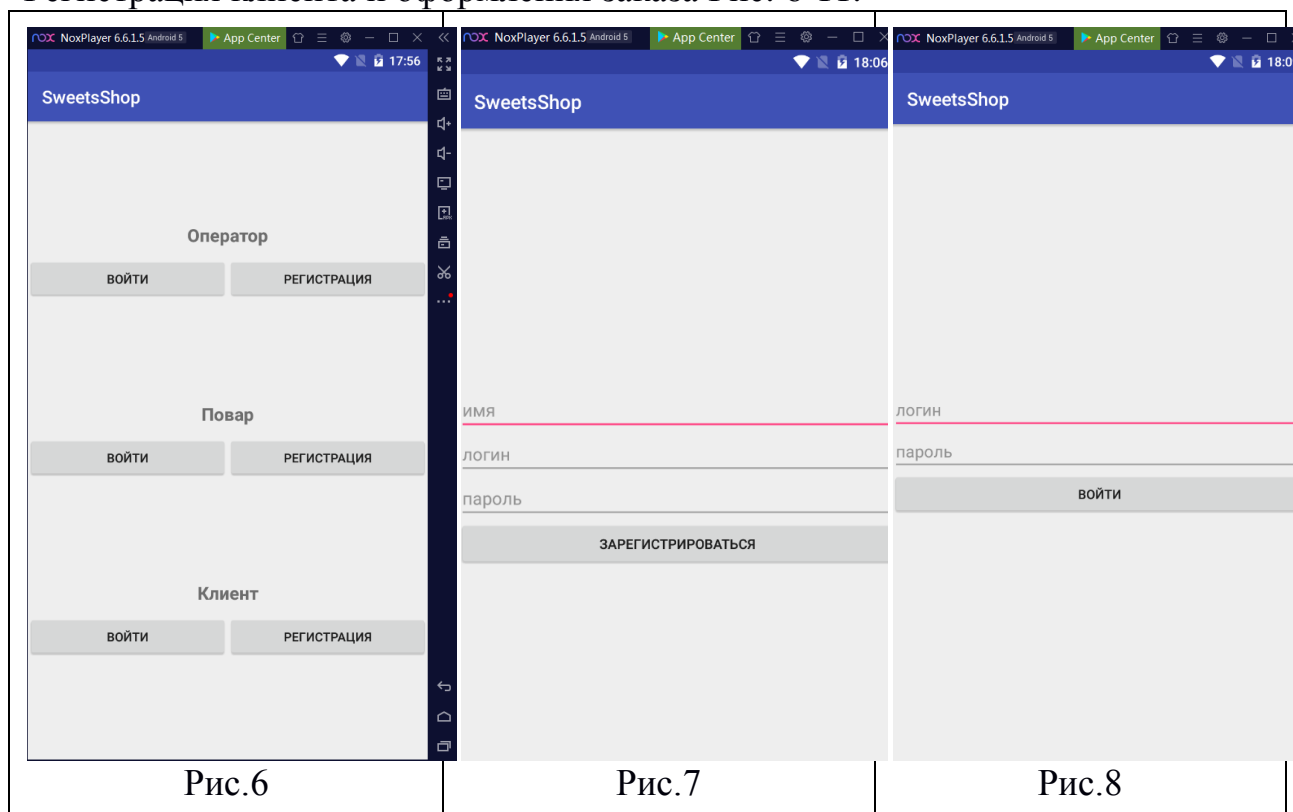
3. Тестирование

Вариант тестирования	Ожидаемый результат	Фактический результат
Регистрация нового пользователя	Добавлен новый пользователь	Совпадает с ожидаемым
	Все поля формы регистрации должны пройти валидацию	Совпадает с ожидаемым
Авторизация	Уведомить, если логин или пароль не прошли валидацию	Совпадает с ожидаемым
Роль Оператора	Оператор может изменять статус заказа	Совпадает с ожидаемым
	Оператор может подтвердить заказ	Совпадает с ожидаемым
Роль Повара	Повар может взять заказы, готовые к выполнению	Совпадает с ожидаемым
Роль Клиента	Клиент может создать заказ	Совпадает с ожидаемым

4. Интерфейс приложения

Интерфейс приложения представлен на рис. 6-8.

Регистрация клиента и оформления заказа Рис. 6-11.



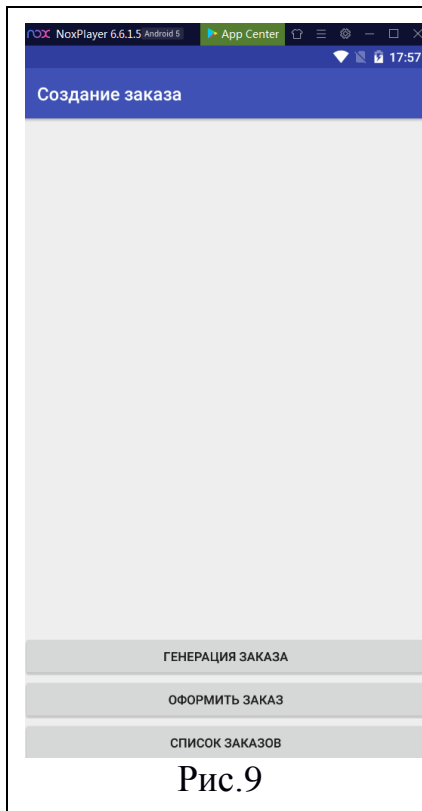


Рис.9

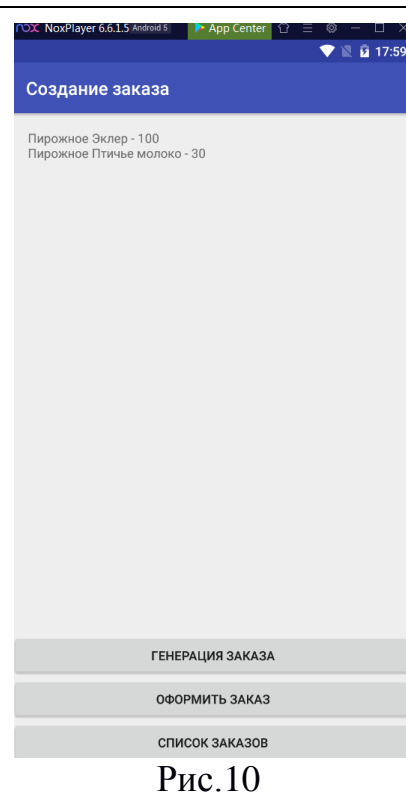


Рис.10



Рис.11

Смена статуса заказа Оператором Рис. 12-14.

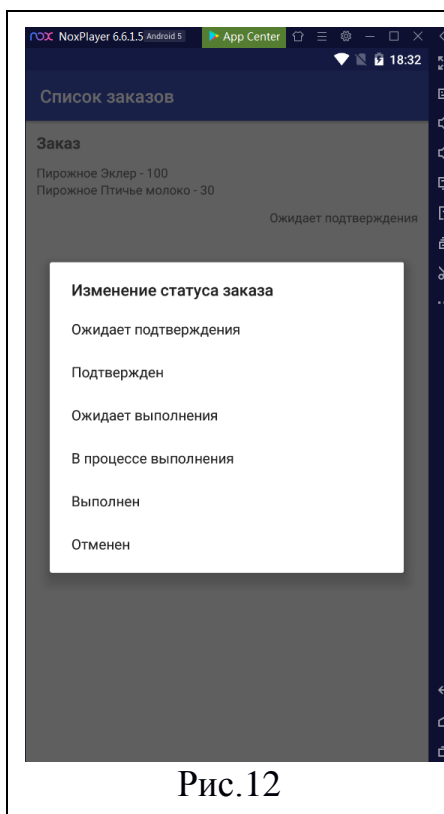


Рис.12

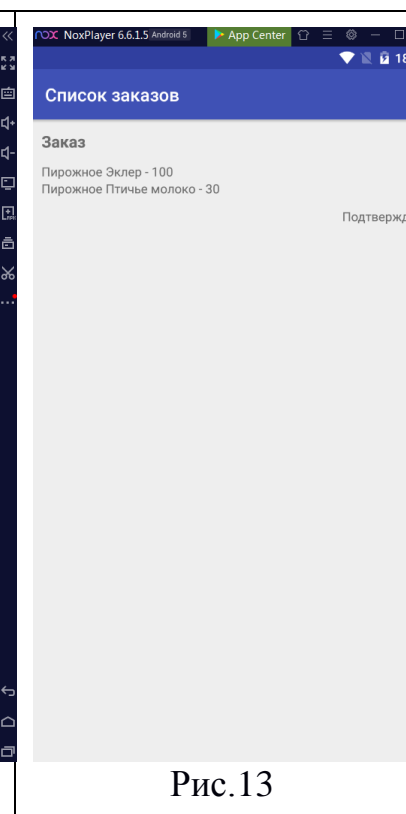


Рис.13

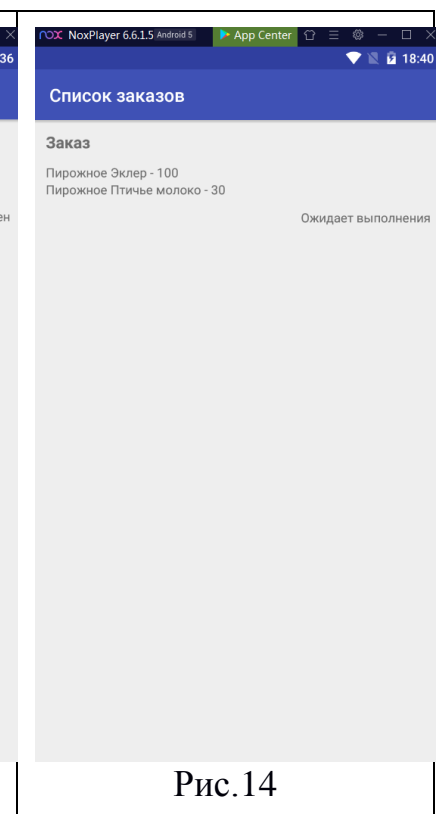
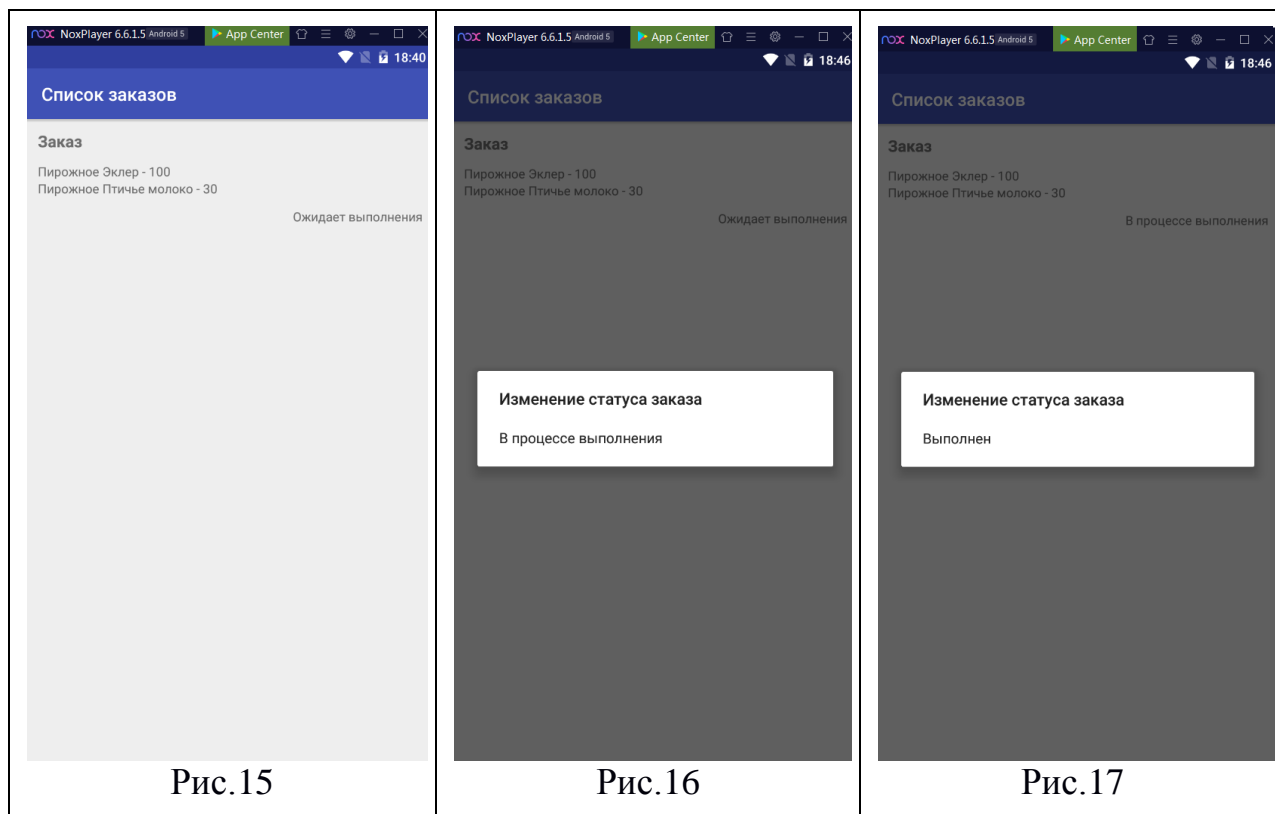


Рис.14

Смена статуса заказа Поваром Рис. 15-17.



5. Инструкция системному администратору по развёртыванию приложения

1. Установить NoxPlayer
<https://ru.bignox.com/>
2. Установить IntelliJ IDEA Ultimate
<https://jetbrains.ru/products/idea/>
3. Скачать архив Sweets_Shop_Client.rar
4. Скачать архив Sweets_Shop.rar
5. Разархивировать Sweets_Shop_Client.rar
6. Разархивировать Sweets_Shop.rar
7. В IntelliJ IDEA Ultimate запустить проект
8. В Multi-Drive Nox создать эмулятор Android 7, перенести в него Sweets_Shop и прописать IP и порт: 192.168.56.1:8080.
9. Прописать в адресной строке хост: localhost:8080/h2-console (рис.18)

Login

Saved Settings: Generic H2 (Embedded) ▼

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password:

Connect Test Connection

Рис.18.

6. Ведение базы данных заказов

Структура БД представлена на рис. 19-20.

Auto commit ☒ Max rows: 1000 Auto complete Off Auto select On

Run Run Selected Auto complete Clear SQL statement:

jdbc:h2:mem:testdb

- CHEF
- CLIENT
- OPERATOR
- ORDERREQUEST
- INFORMATION_SCHEMA
- Sequences
- Users
- H2 1.4.197 (2018-03-18)

SELECT * FROM ORDERREQUEST;

ID	PRODUCTS	STATUS	CHEF_ID	CLIENT_ID	OPERATOR_ID
1	aced0005737200136a6176612e7574696c2...	4	null	1	null
2	aced0005737200136a6176612e7574696c2...	3	null	1	null

(2 rows, 4 ms)

Edit

Рис.19

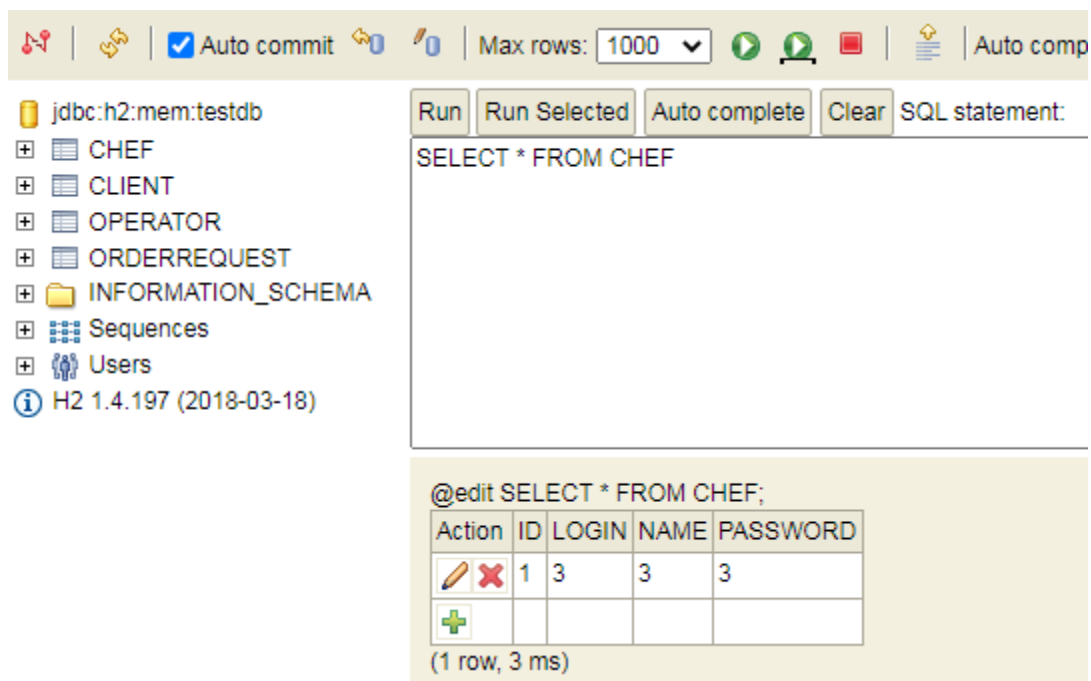


Рис.20

7. Инструкция пользователю по запуску приложения

1. Зарегистрироваться в системе
2. Выбрать понравившийся заказ (для простоты демонстрации реализован генератор заказов)
3. Оформить заказ
4. Дождаться выполнения заказа

Вывод

В результате работы на курсовым проектом было спроектировано и разработано приложение «Sweets Shop». В процессе проектирования были закреплены на практике знания о Spring Framework.

В текущей реализации продемонстрированы такие фундаментальные навыки как построение клиент-серверной системы, организация взаимодействия компонентов.

Код программы выложен в репозитории на GitHub.

Литература

1. Фаулер М. UML. Основы, 3-е издание. Символ-Плюс, 2006.
2. Гранд М. Шаблоны проектирования в Java. BHV-СПб, 2004.
3. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования, 3-е издание. Вильямс, 2007.
4. Гайд по разработке приложений на Spring - <https://spring.io/guides>