Egor Adonev M21-502

Разработка cистемы управления заказами в ресторане

РАЗРАБОТКА ПО, JAVA11, JDK, СРЕДА РАЗРАБОТКИ INTELLIJ, VAADIN,ВЕБ-ИНТЕРФЕЙС,АРМ, РЕПОЗИТОРИЙ, CRM,SPRING BOOT,SPRING DATA JPA, JAVAPERSISTENCYAPI,ПРОВЕРКА ДАННЫХ.

Целью работы является разработка системы управления заказами используя заданный набор технологий.

Для достижения поставленной цели необходимо решить следующие задачи:

* Проектирование БД
* Конфигурация приложения
* View: формы и представления (страниц) UI
* Model: сущности(таблицы) с объявлением полей(столбцов) ограничений в пакете entity
* Controller(service): сервис методов для операций CRUD в пакете service
* Repository: репозитории к сущностям и методы поиска для них в пакете repository
* Разработка программного кода(классов, интерфейсов)
* Развертывание и запуск

Приводятся визуальные результаты работы ПО с помощью Java 11, Vaadin, Spring Boot,Spring Data JPA, Java Persistency API, .

Структура

1 Модель данных

1.1 Сущности(таблицы)

1.2 ORM(Hibernate ORM)

1.3 Java Persistency API

2 Модель UI

2.1 Формы редактирования записей

2.2 Представления таблиц в UI

# 1 Модель данных

# 1.1 Сущности

# 

Таблица Customer:

Она находится в 1NF, так как все атрибуты являются атомарными.

Для 2NF существует полная функциональная зависимость Customer\_Name,

Customer\_Email и Customer\_Phone в первичном ключе Customer\_ID. Это не простые атрибуты, которые полностью функционально зависят от первичного ключа Customer\_ID, как показано ниже.

Customer\_ID🡪 email {полная функциональная зависимость}

Customer\_ID🡪 name{полная функциональная зависимость}

Таблица находится в 3NF, поскольку нет транзитивных зависимостей от первичного ключа.

Таблица Order:

Также находится в 1NF, так как все атрибуты являются атомарными.

Для 2NF существует полная функциональная зависимость Order\_Datetime и Order\_Status от первичного ключа Order\_ID. Это не простые атрибуты, которые полностью функционально зависят от первичного ключа Order\_ID, как показано ниже.

Order\_ID🡪 order\_date(datatype – LocalDateTime) {полная функциональная зависимость}

Order\_ID🡪 ready {полная функциональная зависимость}

Order\_Id 🡪 order\_details {полная функциональная зависимость}

Таблица находится в 3NF, поскольку нет транзитивных зависимостей от первичного ключа.

Таблица Order Detail:

1NF – все атрибуты являются атомарными.

Для 2NF существует полная функциональная зависимость quantity и total\_price от составных ключей, как показано ниже.

order\_ID, food\_item\_ID -> quantity {полная функциональная зависимость}

order\_ID, food\_item\_ID -> total\_price {полная функциональная зависимость}

Таблица находится в 3NF, поскольку нет транзитивных зависимостей от первичного ключа.

Взаимодействие с СУБД происходит за счет репозиториев Spring Data JPA(annotation @Repository) и аннотаций JPA,также в зависимостях – Hibernate логика

Модели таблиц описаны в пакете data.entity

Интерфейсы репо в пакете data.repository

Методы CRUD над записями использующие JPA Repository в data.service

Сервер работает по 3 REST routes (“/”,”/customers”,”/menu”), то есть перенаправляет на нужное представление в соотв. GET запросом

Model - data.entity

View- views

Controller - data.service

# БД – postgreSQL

Серверная часть – Java 11(Spring Boot,Spring Data JPA,Hibernate ORM/JPA)

Клиентская часть – Vaadin

Функции

* Пополнение записей клиентской базы, их удаление и редактирование (управление клиентами)

удаление, редактирование и добавление клиентов с помощью интерфейса репозитория JPA CustomerRepository в котором есть встроенные методы JpaRepository для сохранения и удаления записей в БД , эти методы вызываются в представлении CustomersView(рис. Ниже) при нажатии соответствующих кнопок Save[validateAndSave,в нем вызов fireEvent(CustomerFormEvent.SaveEvent],Delete[CustomerFormEvent.DeleteEvent] и при нажатии на запись вызывается метод editCustomer в котором вызывается метод CustomerForm.setCustomer в котором считывается переданный экземплярИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

* пополнение и редактирование меню

удаление, редактирование и добавление позиций меню с помощью интерфейса репозитория JPA FoodItemRepository в котором есть встроенные методы JpaRepository для сохранения и удаления записей в БД , эти методы(saveFoodItem,deleteFoodItem) вызываются в представлении FoodItemView при нажатии соответствующих кнопок Save[validateAndSave и в нём вызов fireEvent(new FoodItemForm.SaveEvent],Delete[FoodItemForm.DeleteEvent] и при нажатии на запись вызывается метод editFoodItem в котором вызывается метод FoodItemForm.setFoodItem в котором считывается переданный экземпляр FoodItem

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

* добавление заказа на определенную дату и управление заказами ресторана

удаление, редактирование и добавление заказов с помощью интерфейса репозитория JPA OrderRepository в котором есть встроенные методы JpaRepository для сохранения и удаления записей в БД , эти методы(saveOrder,deleteOrder) вызываются в представлении OrderView при нажатии соответствующих кнопок Save( validateAndSave и в нем fireEvent(OrdersFormEvent.SaveEvent) ) ,Delete(OrdersFormEvent.DeleteEvent) и при нажатии на запись вызывается метод editOrder в котором вызывается метод OrderForm.setOrder в котором считывается переданный экземпляр Order

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

JPA Annotations

@Entity – сущность

@Table- таблица

@NotNull – не пустое поле (для типов String и остальных ссылочных типов)

@Id – поле идентификатора (аналог primary key)

@Column – объявления поля

@SequenceGenerator(name = "customer\_id\_sequence", sequenceName = "customer\_id\_sequence", allocationSize = 1) генератор последовательностей ID

@GeneratedValue- данное поле генерируется автоматически

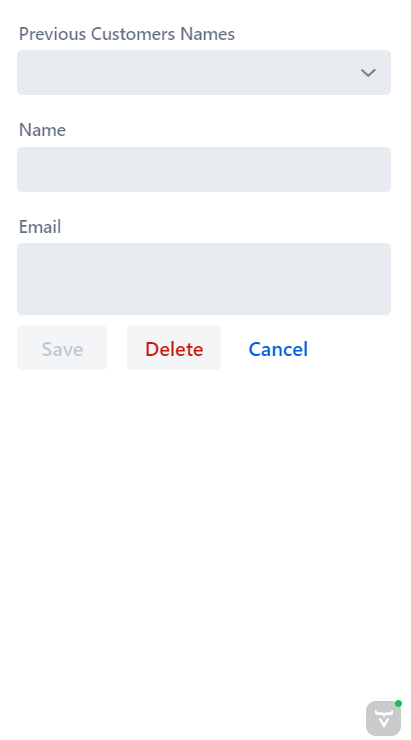


Рис 1 Форма добавления записи в customers

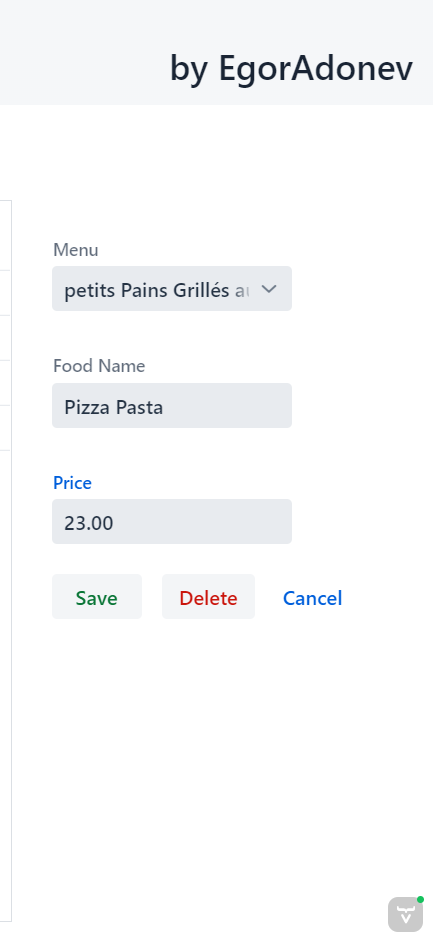


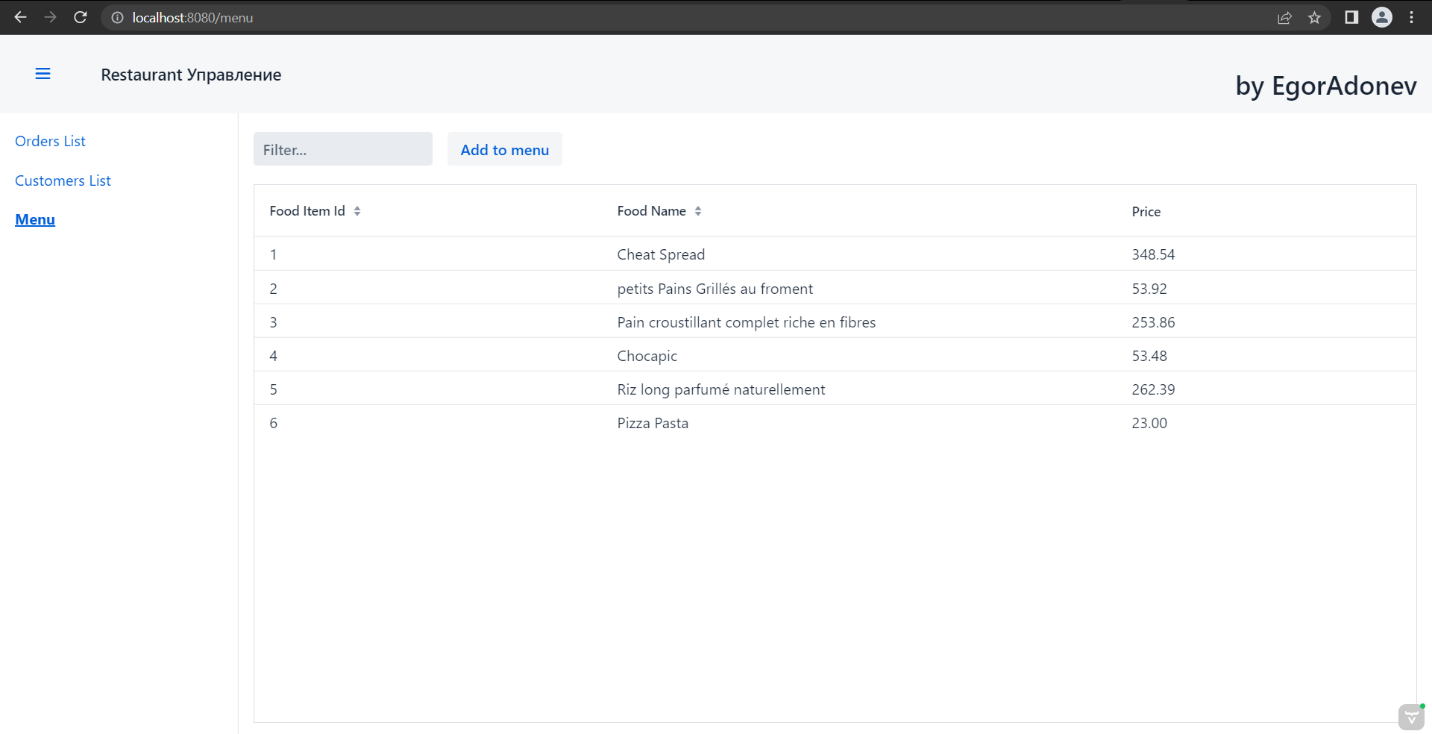
Рис 2 Форма добавления записи в food\_items

Рис 3 Представление FoodItemView

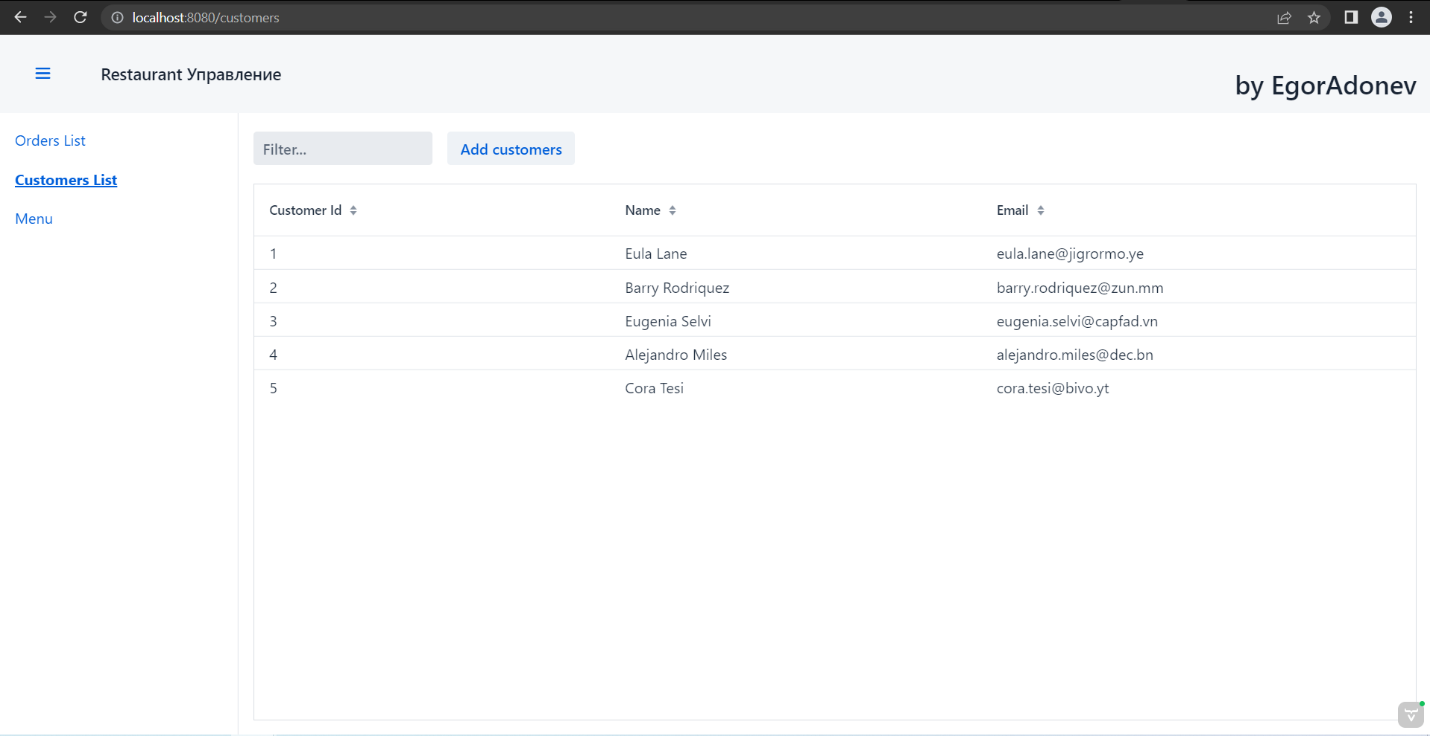


Рис 4 Представление CustomersView

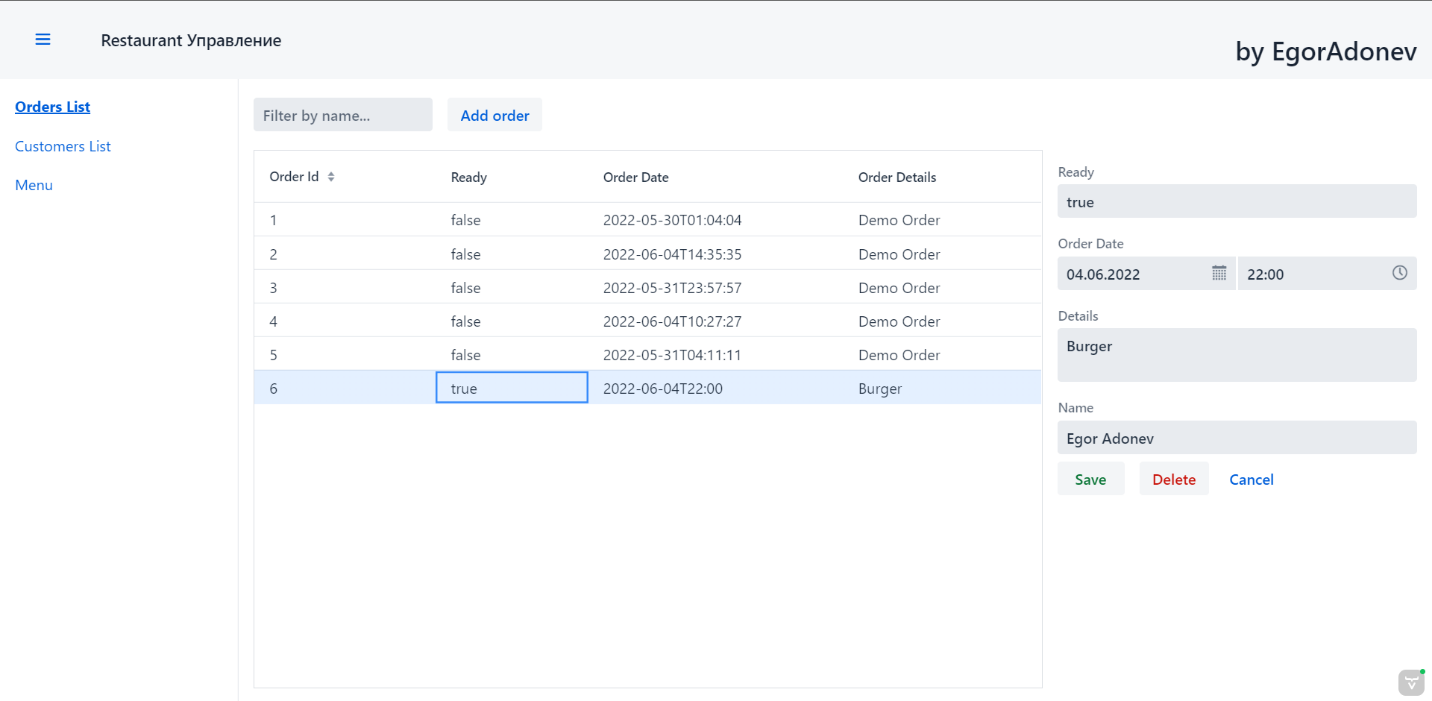


Рис 5 Представление orders с формой на главное странице