Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной

инженерии

**Второй этап курсовой работы**

по дисциплине «Распределенные программные системы»

«Структура бизнес-логики и веб-интерфейса системы.»

**Выполнил:**

Ст. гр. ИСТ-113

Алексеев С.О.

**Принял:**

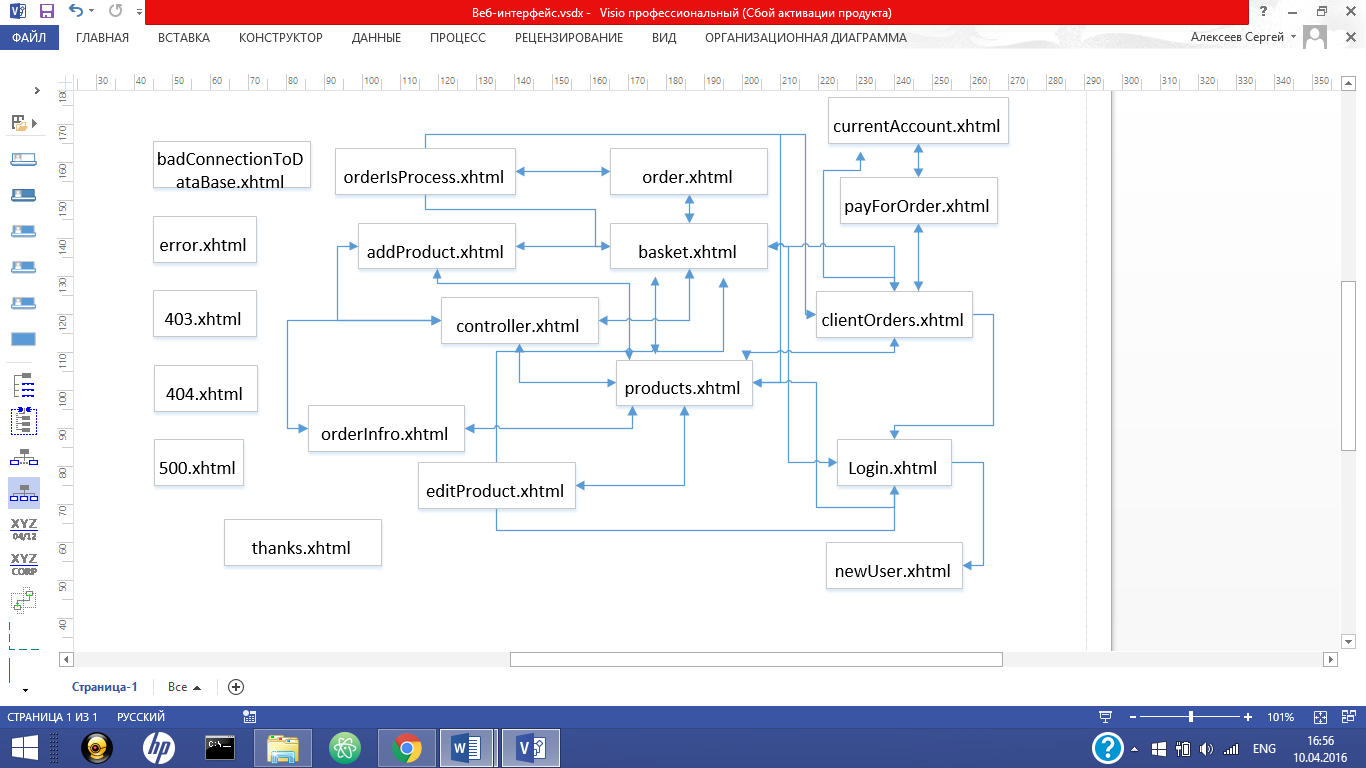
Тимофеев А. А.

Владимир 2016

Схема веб интерфейса системы

Современные концепции проектирования распределенных информационных систем корпоративного уровня предполагают общение пользователя системы с базой данных посредством браузера и технологий Internet. Для обеспечения возможности взаимодействия пользователя с браузером применяется Веб-интерфейс — это совокупность средств, при помощи которых пользователь взаимодействует с веб-сайтом или любым другим приложением через браузер.

Веб-интерфейс моего курсового проекта «Магазин компьютерной техники» приведен на рисунке 1.

Рисунок1. Схема веб-интерфейса системы

**Описание веб-страниц.**

1. JSF-страница thanks.xhtml - в качестве благодарности за покупку создана данная страница.

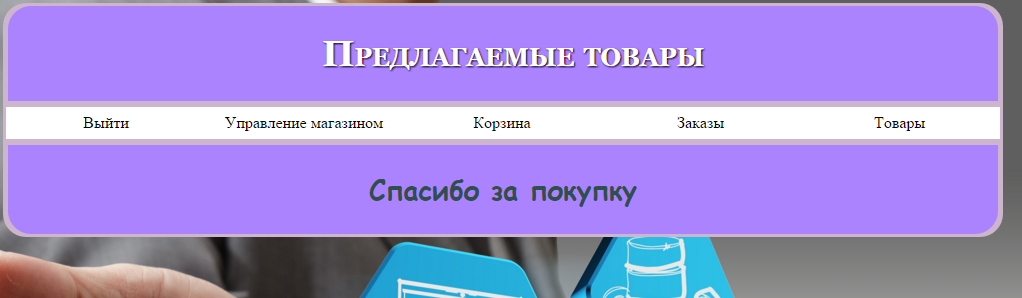


Рисунок 2. Thanks.xhtml

1. JSF-страница products.xhtml – приветствие пользователя и вывод списка товаров. Скриншот выполнен от лица клиента. Продавец может совершать CRUD-операции над товарами.

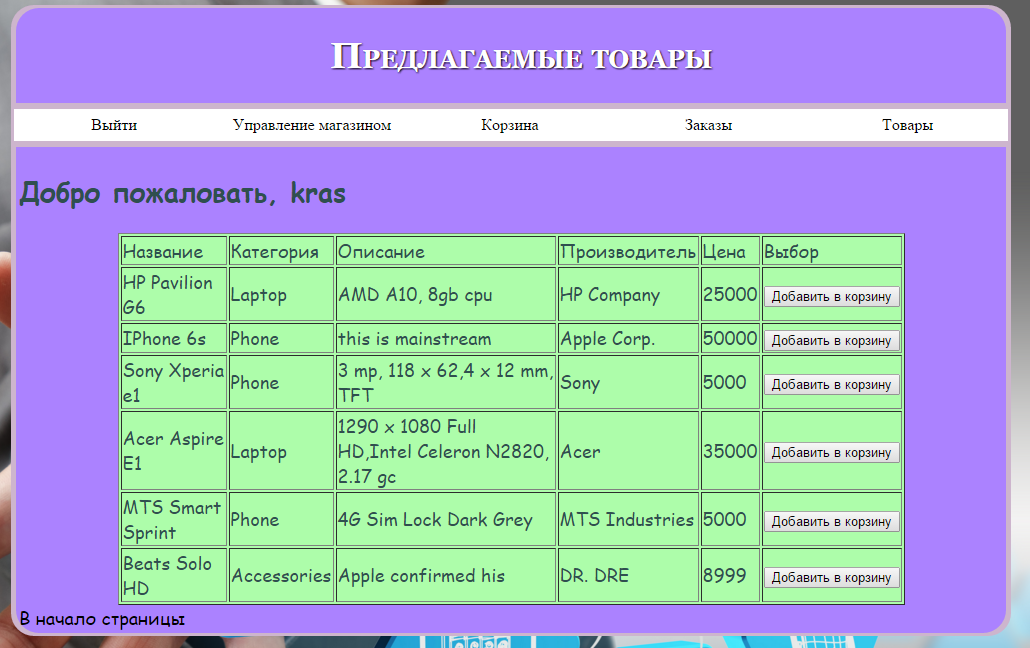


Рисунок 3. Products.xhtml

1. JSF-страница orderIsProcess.xhtml – окно для заполнения данных получателя, если клиент выбрал доставку на дом.

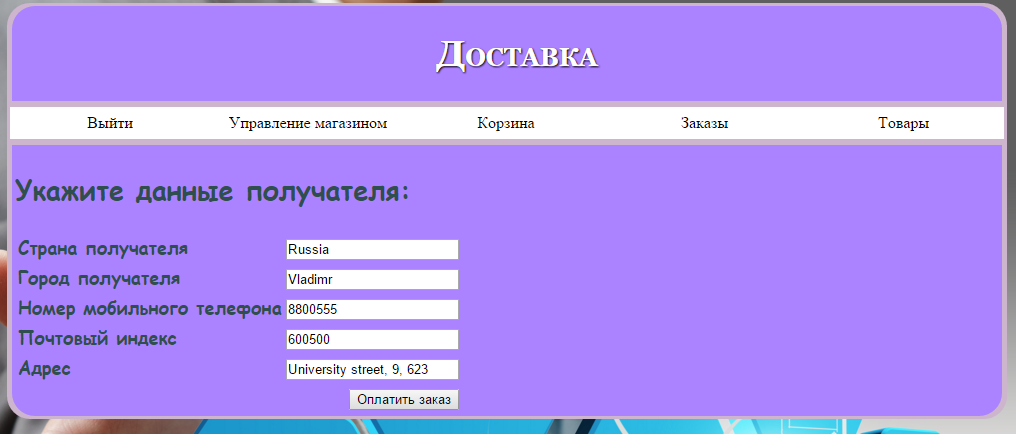


Рисунок 4. orderIsProcess.xhtml

1. JSF-страница orderInfo.xhtml – данная страница будет отображать заказы клиентов. Продавец может изменить состояние заказа. В дальнейшем планируется добавление поиска заказов по имени клиента. И добавление атрибута «Итоговая цена».

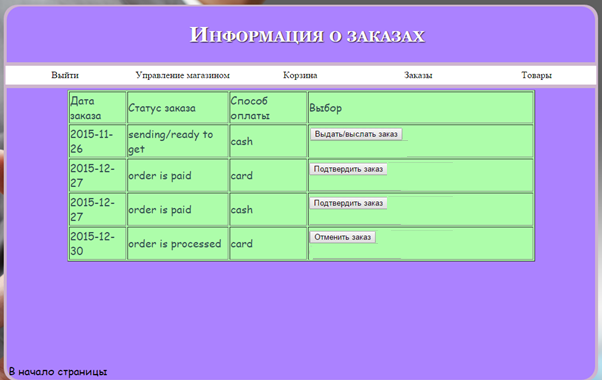


Рисунок 5. orderInfo.xhtml

1. JSF-страница order.xhtml – оформление заказа.

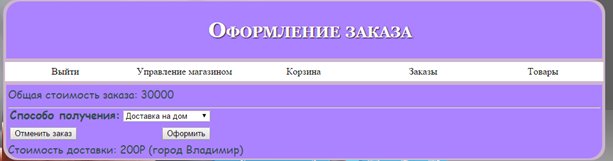


Рисунок 6. order.xhtml

1. JSF-страница payForOrder.xhtml – оплата заказа.



Рисунок 7. order.xhtml

1. JSF-страница newUser.xhtml – регистрация пользователя в системе. По умолчанию присваивается роль клиента.

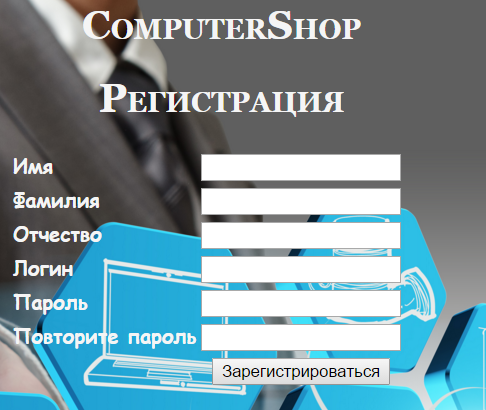


Рисунок 7. newUser.xhtml

1. JSF-страница login.xhtml – вход в систему.



Рисунок 8. Login.xhtml

1. JSF-страница error.xhtml – произошла какая-либо ошибка.

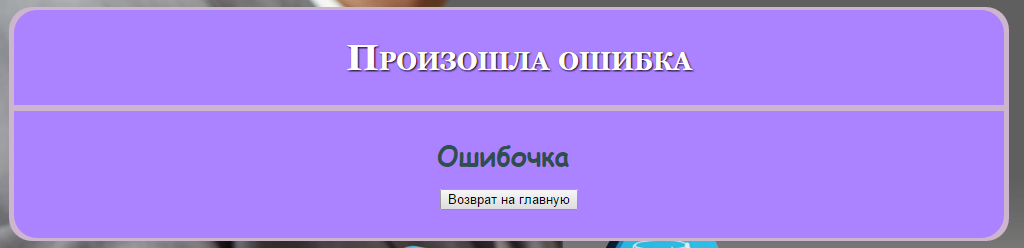


Рисунок 9. Error.xhtml

1. JSF-страница editProduct.xhtml

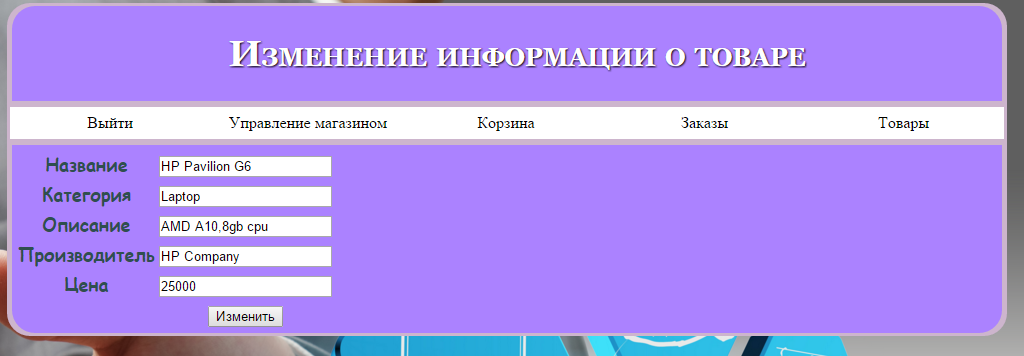


Рисунок 10. editProduct.xhtml

1. JSF-страница controller.xhtml – основной функционал работников магазина.

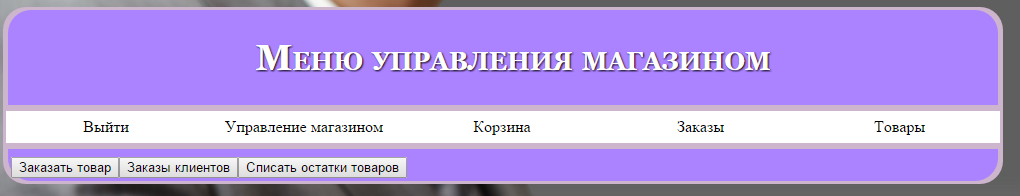


Рисунок 11. Controller.xhtml

1. JSF-страница clientOrders.xhtml – история покупок клиента.

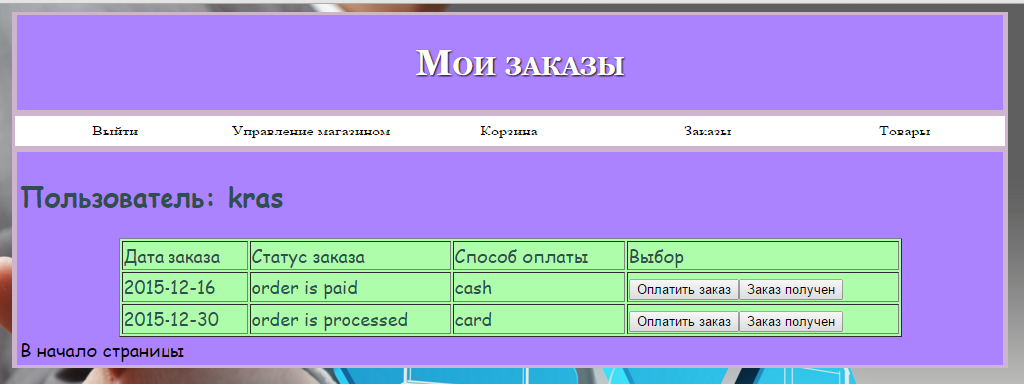


Рисунок 12. clientOrders..xhtml

1. JSF-страница basket.xhtml – корзина клиента.

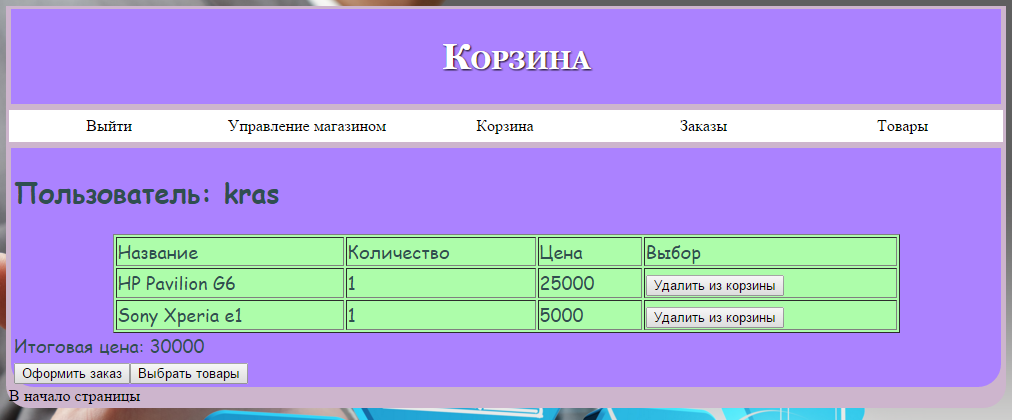


Рисунок 13. basket.xhtml

1. JSF-страница badConnectionToDataBase.xhtml - страница ошибок, связанная с БД.
2. JSF-страница addProduct.xhtml – заказ товара в магазин.

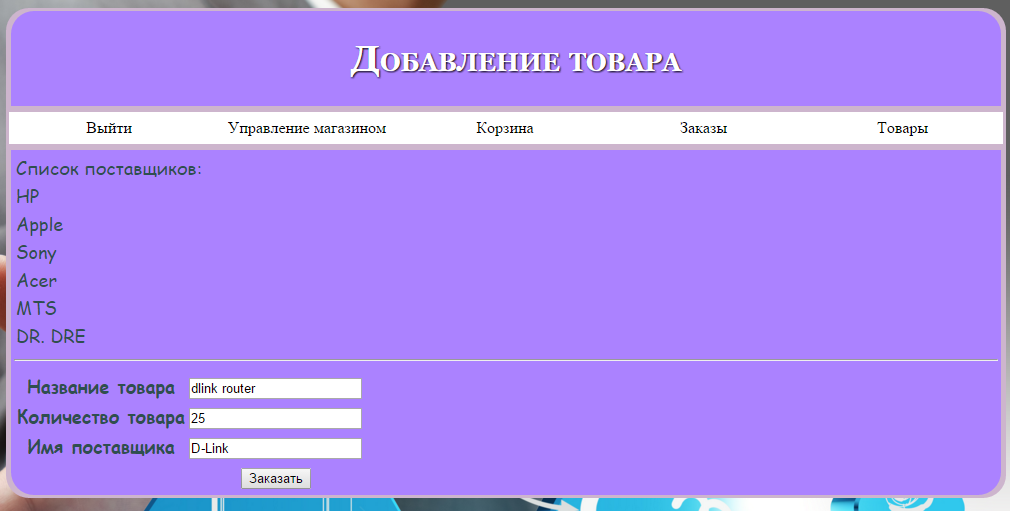


Рисунок 14. addProduct.xhtml

1. JSF-страница 500.xhtml
2. JSF-страница 404.xhtml
3. JSF-страница 403.xhtml
4. JSF-страница currentAccount.xhtml – страница личного счета клиента. Здесь он может его пополнить, либо посмотреть историю покупок. Если пользователь – администратор, то на странице будет показан оборот продаж и капитал магазина.

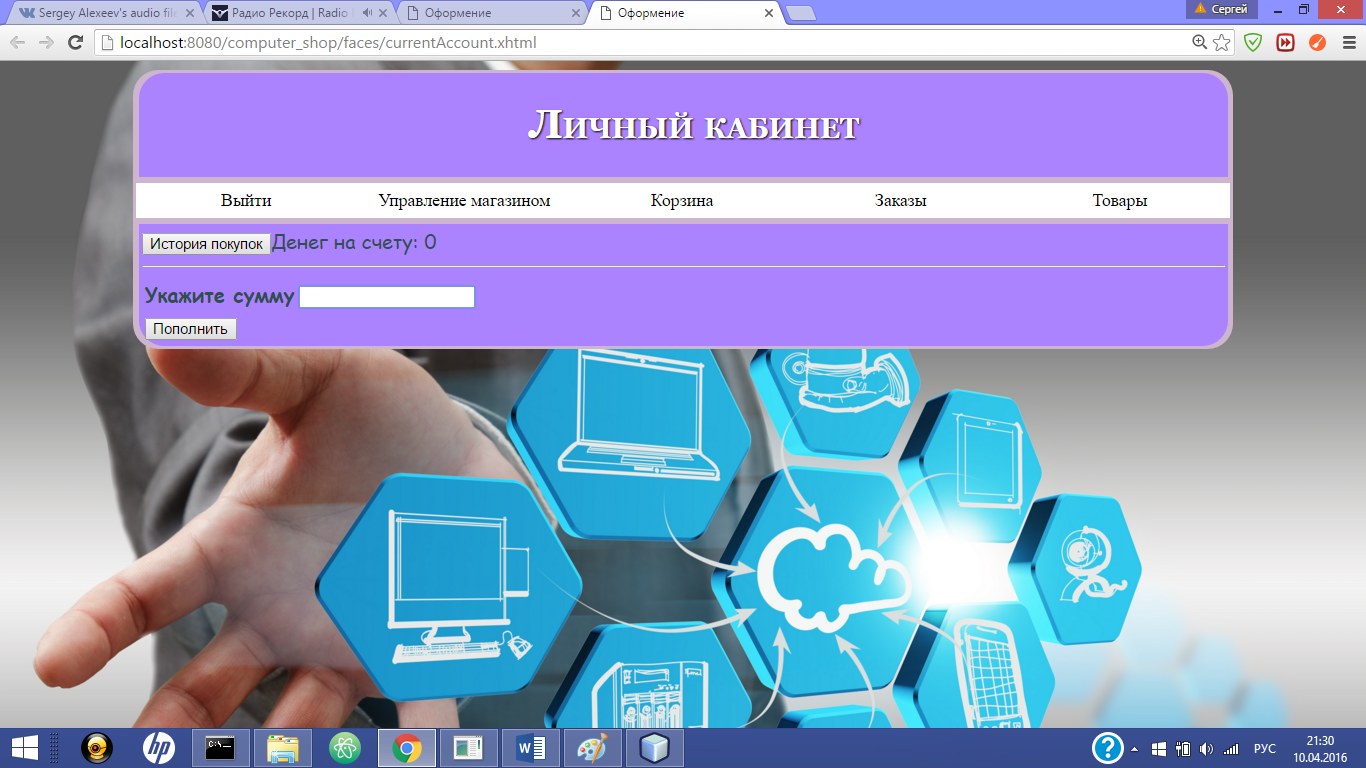


Рисунок 15. currentAccount.xhtml

В ходе разработки системы, будут происходить изменения в дизайне и функционале.

Диаграмма классов сущностей.

Диаграммы классов показывают статическую структуру системы, то есть определяют типы объектов системы и различного рода статические связи и отношения между ними. Диаграммы классов содержат набор статических элементов, как, например, классы, типы, их связи, объединенные в граф.

Диаграмма классов сущностей информационной системы «Магазин компьютерной техники» приведена на рисунке 16.

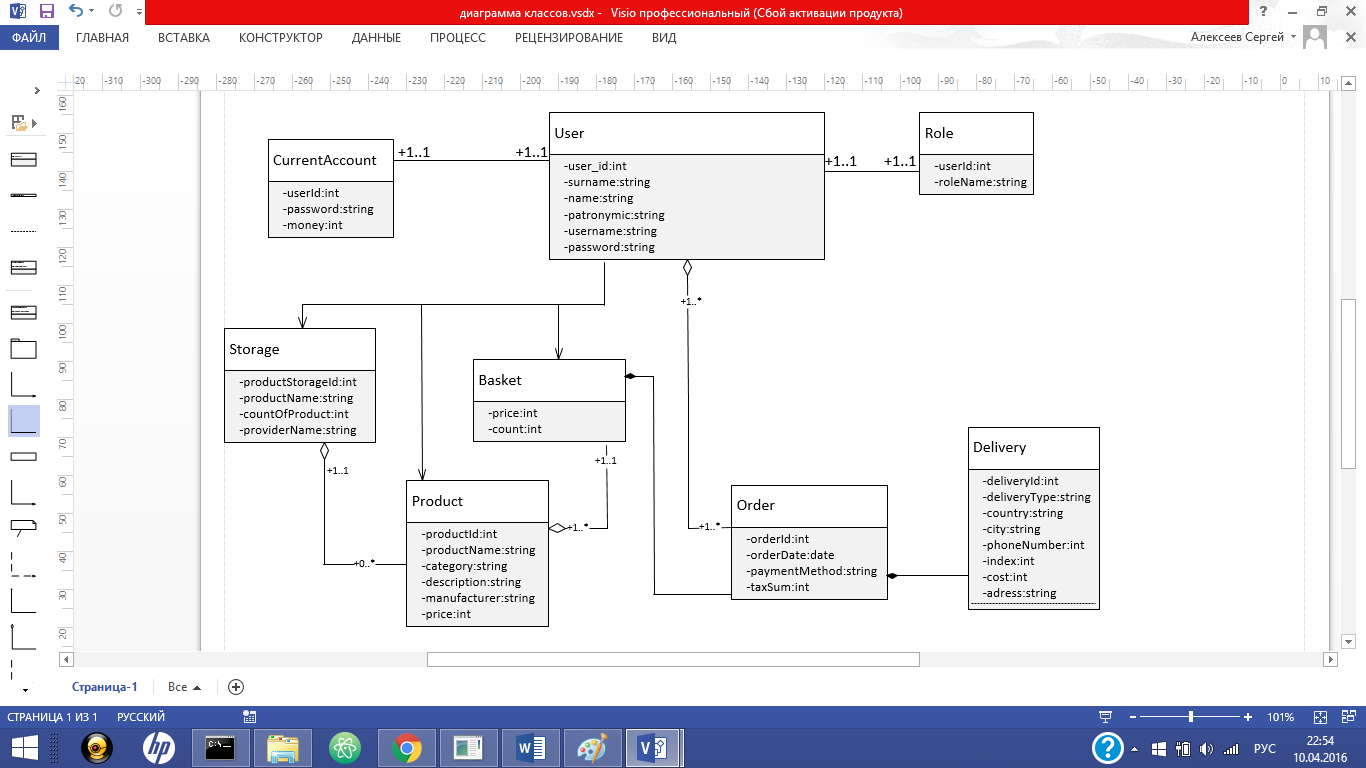


Рисунок16. Диаграмма классов сущностей.

**Описание классов.**

Пользователь (User) – содержит информацию о пользователе.

Атрибуты:

* userName – имя пользователя.
* password – пароль пользователя.
* userId – номер пользователя.
* Name - Имя пользователя.
* Surname - Фамилия пользователя.
* Patronymic - Отчество пользователя.

Роль (Role) – роль пользователя.

Атрибуты:

* userId – номер пользователя.
* roleName – название роли.

Лицевой счет (CurrentAccount) – информация о счете пользователя. У администратора свой счет, который означает капитал магазина.

Атрибуты:

* userId – номер пользователя.
* password – дополнительный пароль.
* Money – количество денежных средств (в рублях).

Товар (Product) – содержит информацию о товарах магазина

Атрибуты:

* productName – название товара
* category – категория товара
* manufacturer – производитель товара
* description – описание товара
* price – цена товара (оптовая)

Корзина (Basket) – содержит информацию о товаре и его количестве.

Атрибуты:

* productID – номер продукта
* count – количество

Заказ (Order) – содержит информацию о заказе.

Атрибуты:

* orderDate – дата совершения заказа.
* paymentMethod – способ оплаты заказа.
* orderStatus – состояние заказа.
* taxSum – итоговая цена заказа.

Доставка (Delivery) – содержит информацию о доставке.

Атрибуты:

* deliveryType – тип получения.
* deliveryStatus – состояние доставки.
* country – страна получателя.
* city – город получателя.
* phoneNumber – номер телефона получателя.
* index – почтовый индекс.
* cost – стоимость доставки.
* adress – адрес доставки.

Storage – информация о заказах на склад.

Атрибуты:

* productId – номер товара.
* countOfProduct – количество товара.
* providerName – имя поставщика.

Диаграмма последовательностей.

**Описание прецедента «Добавить товар в корзину»**

Расширенное описание прецедента «Добавить товар в корзину»

Название: «Добавить товар в корзину»

Предусловие: Клиент должен войти в систему.

Действующее лицо: Клиент

Основной поток: Заполнение корзины

Клиент открывает страницу (окно приложения), отображающую каталог товаров, предоставляемых компьютерным магазином. Клиент добавляет необходимые товары в корзину. После этого он получает возможность оформить заказ.

Альтернативный поток: Данного товара нет в каталоге. Клиент изменяет выбор.

Постусловие: Если товары есть в каталоге, то клиент получает право оформить заказ.

Диаграмма последовательности является одной из разновидности диаграмм взаимодействия и предназначена для моделирования взаимодействия объектов Системы во времени, а также обмена сообщениями между ними.

Диаграмма последовательностей прецедента «Добавить товар в корзину» приведена на рисунке 17.

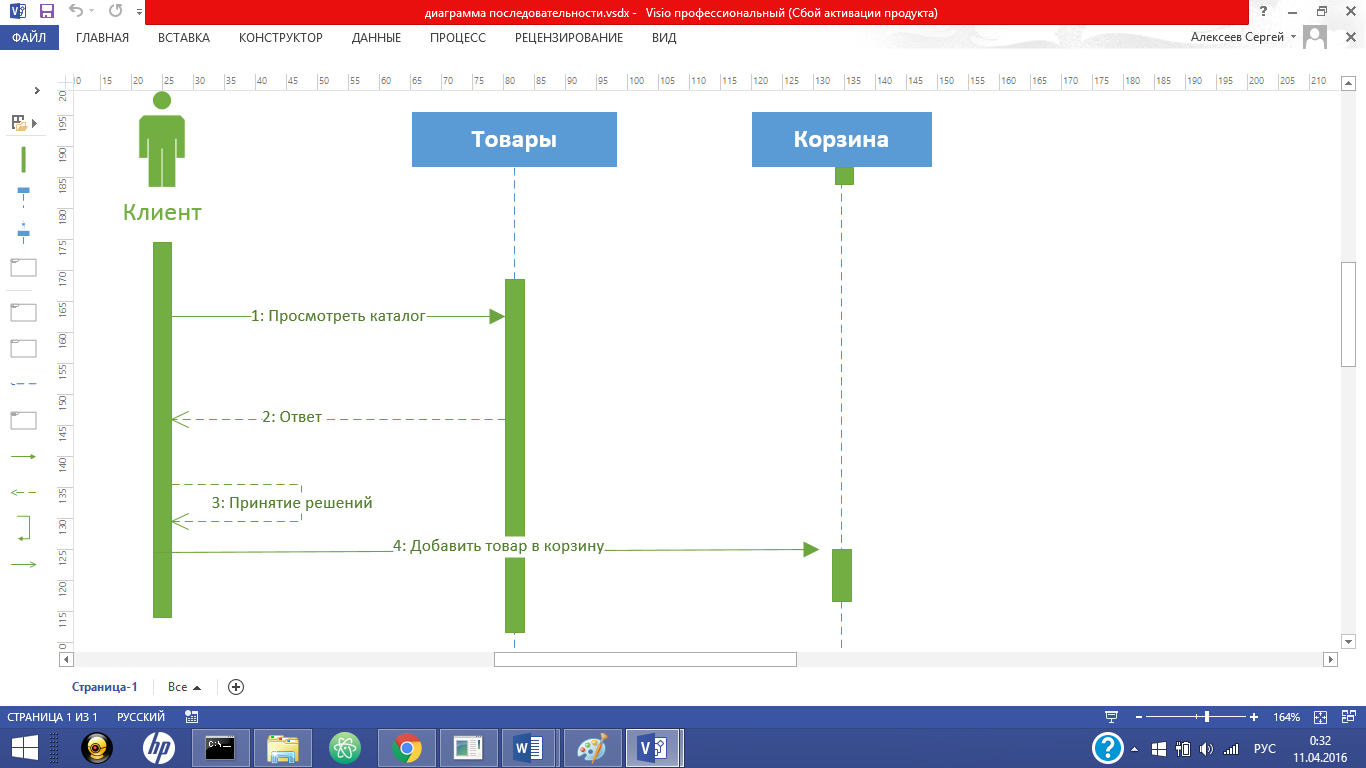


Рисунок17. Диаграмма последовательности для прецедента «Добавить товар в корзину»

 Сводная диаграмма классов/компонентов приложения с указанием связей между компонентами различных слоев.

Слой представления - это единственный слой, непосредственно связанный с пользователем. Слой представления используется для получения данных от пользователя и передачи их слою бизнес-логики для дальнейшей обработки, и когда данные получены в объекте-значении, он отвечает за представление объекта-значения в надлежащей форме, которая понятна пользователю.

Слой бизнес-логики (BUS) работает как мост между уровнем представления и DAO. Все значения пользователя, полученные от уровня представления, передаются в BUS. Слой бизнес-логики содержит всю бизнес-логику программы.

Слой доступа к базе данных (DAO) строит запрос на основе параметров, полученных от слоя бизнес-логики и используя соединение с базой данных, выполняет требуемые операции. Затем происходит возврат результатов из DAO классов в слой бизнес-логики.

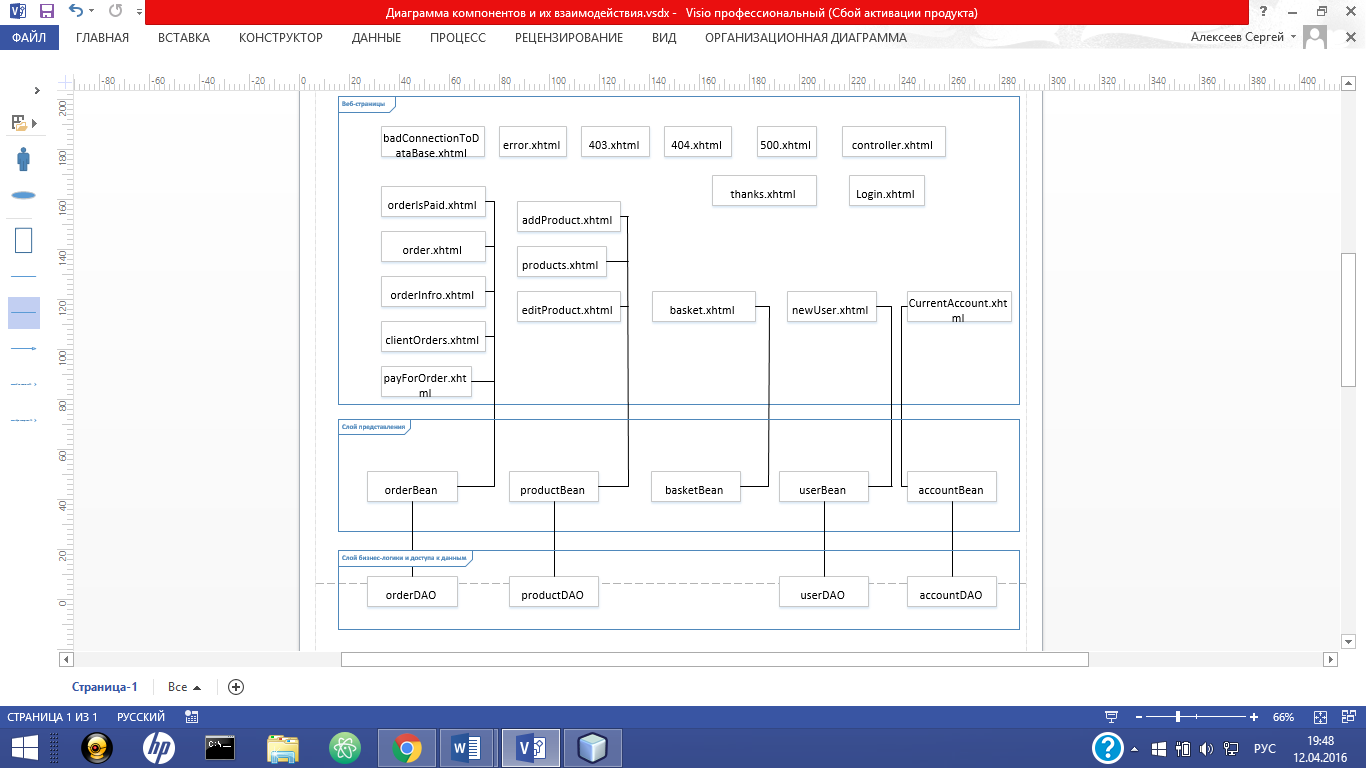


Рисунок18. Диаграмма компонентов и их взаимодействия

Спецификация компонентов системы.

**productDAO:**

* addToCatalog(productName, category, description, manufacturer, price) – добавление товара в каталог.
* deleteFromCatalog(productId) – удаление товара из каталога.
* updateProductInformation(productId, productName, category, description, manufacturer, price) – обновление информации о товаре.
* showCatalog() – посмотреть все товары.
* searchProduct(productName) – поиск товаров по названию.
* getProvidesNames() – просмотр справочника поставщиков.
* orderTheProduct(productName,count,provider) – заказать товар.
* deleteProductsFromStorage() – списать остатки товаров со склада.
* addProductToStorage(productStorageId) – добавить товар на склад.
* deleteProductFromStorage(productStorageID) – удалить товар со склада.

**orderDAO:**

* showOrdersInformation() – посмотреть информацию о заказах.
* cancelOrder(orderId) – отменить заказ.
* confirmOrder(orderId) – подтвердть заказ.
* issueTheProducts(orderId) – выдать/выслать товары.
* processOrder(paymentMethod,userID) – оформить заказ.
* payForOrder(orderID,userID) – оплатить заказ.
* feedback(orderID) – подтвердить получение.
* setDelivery(userID,deliveryType,country,city,phoneNumber,postcode,cost,address) – указание данных для доставки заказа.
* showOrderInformation(userId) – посмотреть информацию о заказах.

**userDAO:**

* saveUser() – сохранение пользователя в базе данных.
* getUserIdByLogin(username) – получение номера пользователя по логину.
* searchUserByLogin(username) – поиск пользователя в базе данных по логину.

**accountDAO:**

* showHistory(userID) – посмотреть историю покупок.
* showBalance(userID) – посмотреть остаток денежных средств.
* addFunds(money, accountID) – пополнить счет.