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

Лабораторная работа №2

по дисциплине "Технологии программирования"

ТЕМА РАБОТЫ:

Динамическое моделирование работы системы

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Принял:

Ассистент кафедры ИСПИ

Данилов В.В.

Владимир 2022 г.

Цель работы:

Изучить методы анализа поведения компонентов программной системы путем синтеза диаграмм, описывающих поведение моделируемой ИС с использованием нотации UML.

Предметная область: Магазин компьютерной техники.

Описание предметной области:

В данной лабораторной работе будет использоваться предметная область розничной компании, занимающейся продажей электронных устройств (ноутбуков, компьютеров, смартфонов) в некоторых магазинах розничной сети, расположенных в черте города (пунктах выдачи заказов электронного сайта сети).

Основные сведения о действиях компании:

-компания осуществляет закупку компьютерной техники по некоторым ценам у производителей

-осуществляет доставку и распространение устройств компьютерной техники в конечные филиалы (магазины розничной сети)

-компания проводит набор кадров на должности продавцов и менеджеров

-компания продает свои модели телефонов клиентам посредством их привлечения (рекламой в интернете, проведением грамотной скидочной политики и тд).

Начнем с анализа состояний в автоматизируемой системе электронного компьютерного магазина для построения UML диаграммы состояний

Сделаем это на примере состояний объекта «Заказ» и «Корзина» для данной системы.

Состояния объекта «Заказ в системе»:

1. Покупатель собрал желаемые товары в корзине, указав их требуемое количество. Объект «Корзина» заполнен. Начинает создаваться объект «Заказ»
2. От системы создания заказа приходит форма для оформления заказа, в которой необходимо заполнить данные о способах оплаты, реквизитах пользователя, место для получения заказа.

А) Заказ не будет создан до того момента, пока не будут заполнены реквизиты и не будет выбрано доступное место получения заказа.

Б) Оплата заказа производится через внешнюю платежную систему, заказ будет обработан только после получения подтверждения о его оплате.

3) От системы проверки оформления заказа пришло сообщение (внутри системы) об отсутствии всех необходимых введенных данных.

Покупателю на сайте показывается соотвествующее уведомление, переходим к пункту 8

4) От платежной системы пришло сообщение об отказе в оплате или истек максимальный срок ожидания оплаты.

Заказ отменяется, пользователю приходит на электронную почту уведомление об отмене заказа. После этого переходим к пункту 8.

1. От платежной системы пришло подтверждение об оплате. Заказ помечается как «оплаченный».

Указанный покупателем адрес доставки проверяется во внешней системе проверки адресов, то есть заказ ожидает подтверждение о проверке адреса.

1. От системы проверки адресов пришло сообщение об ошибке в указанном для доставки адресе. Покупателю приходит уведомление на почту об отмене заказа и возврате средств, платежная система получает команду вернуть деньги за отмененный заказ на счет пользовтеля.
2. От системы проверки адресов пришло подтверждение правильности введенного адреса. Заказ помечается как «доставляемый» и переходит в состояние ожидания подтверждения о доставке.

Пользовтелю приходит уведомление на почту о передаче заказа в службу доставки.

1. Заказ помечается как «не созданный» и «ждёт» дооформления.
2. Заказ помечается как «отмененный» (и «ждёт» переоформления)
3. Заказ приходит в пункт выдачи и помечается как «досталенный». Покупатель получает соответствующее уведомление о доставке товара.

Диаграмма состояний (Statechart Daiagram UML):

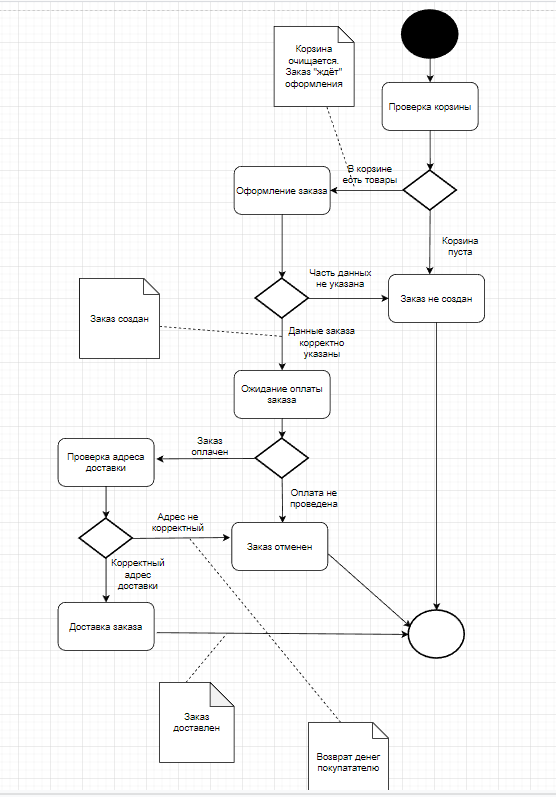


Рисунок 1. Диаграмма состояний системы

Далее проанализируем виды деятельности в системе и построим диаграмму последовательностей.

Используем для это логику перехода между состояниями объекта «Заказ» из построенной и описанной ранее диаграммы состояний.

Выделим такие основные последовательности (последовательные «остановки» для создаваемого заказа в системе, вызываемые функции и сопровождаемые ими действия, а также ответные реакции системы (тоже в виде вызова функций).

Последовательные «звенья» системы оформления заказа:

- Покупатель

Действия (функциональные возможности – прецеденты):

- Добавить товар в корзину: ответное действие – сообщение о добавлении товара от системы

- Открыть страницу «Корзина»: ответное действие – показать текущее состояние корзины на новой странице электронного сайта магазина

- Создать заказ: ответное действие – показать статус созданного заказа и перейти к следующим шагам подтверждения оформления заказа (заказ может иметь на этом этапе состояния «Создан» или «Не создан» (по причине указания неполноты данных или их некорректности)

- Оплатить заказ: ответное действие – показать состояние оплаты заказа (заказ на данном этапе может иметь состояния «Оплачен» или «Не оплачен»)

- Profile Controller (требуется для регистрации пользователей в системе, проверке данных, вводимых пользователем при его авторизации в системе)

- BasketManager

Использует дополнительное звено (объект Basket)

Basket (объект «Корзина», над которым осуществляются вышеперечисленные действия и вызываются функции с помощью звена BasketManager)

Функции:

- AddProductToBasket(): Product (using Basket)

Ответное действие системы – ShowProductPrompt()

- ChangeProductQuantity(): Product (Изменение кол-ва одного из товаров в корзине – using Basket)

Ответное действие – ChangeBasketConditionList() : Basket

- OrderManager

Используемые «подзвенья» звеном OrderManager:

- Order

- PaymentSystem

- CheckingAdressSystem

Функции:

- CreateOrder(): Order

Ответное действие системы - ShowOrderInfo()

- SetOrderStatus() : Order

Ответное действие системы – GetConfirmationOfOrder()

- PayOrder() : Order (using Payment System)

Ответное действие системы – GetPaymentStatus()

- CheckAdress() : Order (using CheckingAdressSystem)

Ответное действие системы – GetAdressExistense()

- CancelPayment() : Order (using Payment System)

Ответное действие системы – ChangeStatusPaymentOrder()

Диаграмма последовательностей (Sequence UML Diagram):

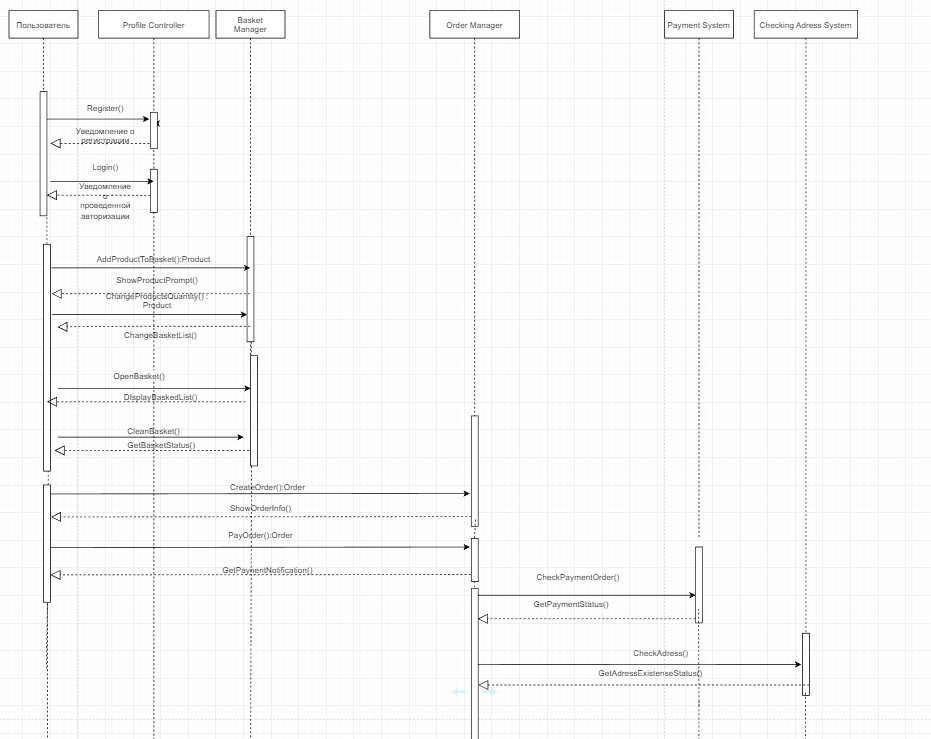


Рисунок 2. Диаграмма последовательностей

Вывод

В результате выполнения работы я изучил методы анализа поведения компонентов программной системы путем синтеза диаграмм, описывающих поведение моделируемой ИС с использованием нотации UML.