

Задание

Необходимо построить объектно-ориентированную модель заданной предметной области, используя язык UML (Unified Modeling Language – унифицированный язык моделирования).

Набор моделей, используемых в объектно-ориентированном подходе, включает:

- диаграммы вариантов использования (Use Case Diagram);
- диаграммы взаимодействия (Interaction Diagram), к которым относятся диаграммы последовательности (Sequence diagram) и диаграммы взаимодействий (Collaboration diagram);
- диаграммы классов (Class Diagram);
- диаграммы состояний (State Machine Diagram);
- диаграммы действий (Activity Diagram);
- диаграммы компонентов (Component Diagram);
- диаграммы размещения (Deployment Diagram).

В данной работе приведен проект построения программного обеспечения информационной системы. В качестве предметной области рассмотрен «Отдел сбыта предприятия». Проектирование производилось с помощью специализированного программного обеспечения Visual Paradigm Standart Edition v15.2 (бесплатная Trial-версия) – программного пакета для создания диаграмм нотации UML.

Отдел сбыта является самостоятельным структурным подразделением предприятия. Основная задача отдела сбыта – реализация продукции, производимой на предприятии.

В данной курсовой работе рассмотрен отдел сбыта предприятия, производящего некоторую продукцию по предварительному заказу. Клиент выбирает нужные товары из каталога, и делает заказ. После готовности заказа ему доставляют товар.

Основные функции отдела сбыта предприятия:

- Продажа товаров:
 - Прием заявок на заказ товаров;
 - Рассмотрение заявок на заказ товаров, оформление ответа;
 - Передача клиентам договоров по заявкам, составленных в юридическом отделе, выставление счетов на оплату;
 - Составление спецификации по заказанным товарам для производственного отдела;

- Организация приема, хранения, комплектации, консервации и упаковки готовой продукции от отдела производства;
 - Организация отгрузки готовой продукции, контроль и учет выполнения планов реализации готовой продукции;
- Ведение базы данных клиентов;
- Контроль над получением оплаты совместно с бухгалтерией;
- Ведение каталога изделий, которые может заказать клиент:
 - Обновление списка товаров;
 - Обновление прайс-листа;
- Рассылка клиентам рекламных сообщений, составленных рекламным отделом;
- Составление плана продаж продукции, установка цен;
- Ведение базы данных сотрудников отдела сбыта:
 - Обновление прав доступа;
 - Обновление списка сотрудников.

Также в курсовой работе затронуты некоторые функции других отделов – юридического, производственного, финансового, бухгалтерского и рекламного отделов. Отделы на предприятии контактируют друг с другом, и поэтому требуется более широкое рассмотрение системы.

4. Проектирование программного обеспечения ИС «Отдел сбыта предприятия»

4.1. Диаграмма вариантов использования

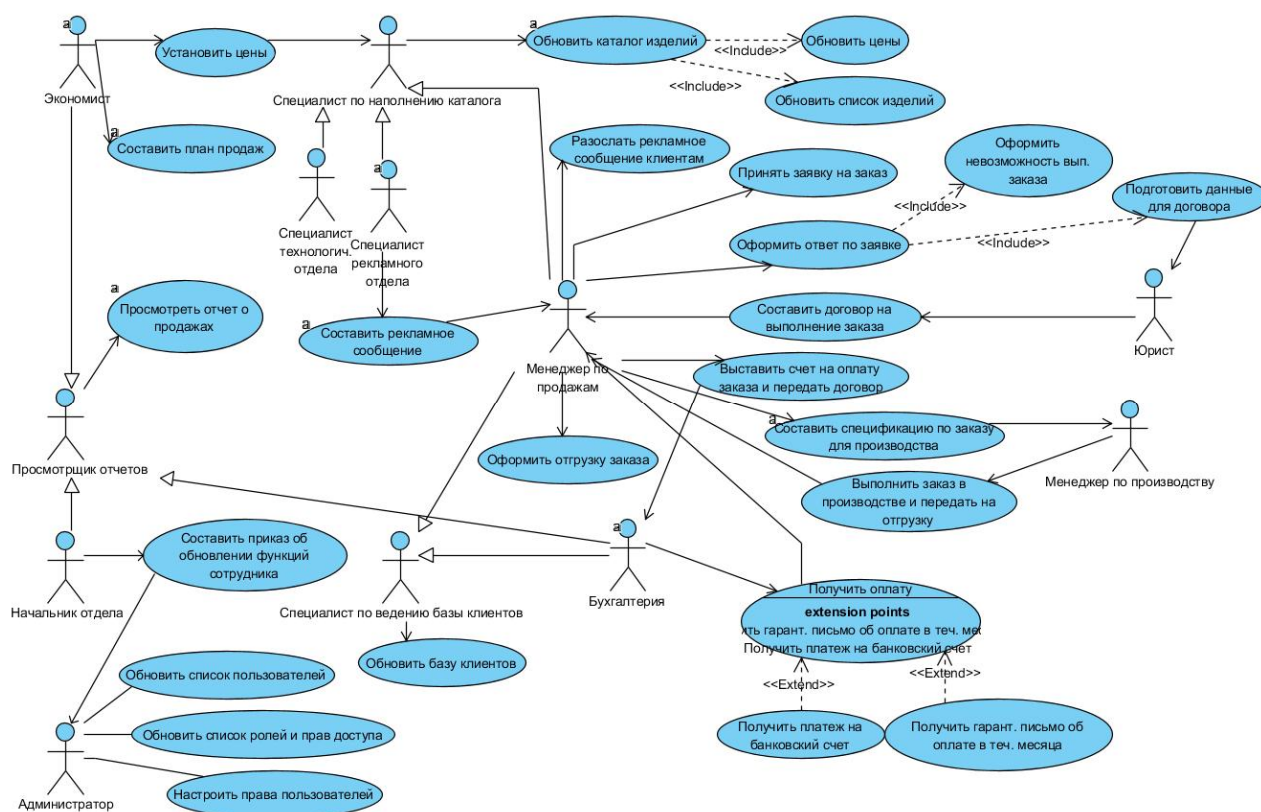


Рисунок 1 - Диаграмма вариантов использования отдела сбыта предприятия

4.1.1. Описание вариантов использования

«Установить цены» – актер «Экономист» устанавливает цены и создает прайс-лист изделий.

«Составить план продаж» – это также делает актер «Экономист» – составление плана продаж на определенный период.

«Посмотреть отчет о продажах» – отчет о продажах могут смотреть несколько акторов: «Экономист», «Начальник отдела», «Бухгалтерия». На диаграмме создан условный (не существующий реально на предприятии) актер «Просмотрщик отчетов» и выполнена генерализация от реальных акторов.

«Составить рекламное сообщение» – занятие для «Специалиста рекламного отдела». Данный актер создает некое рекламное сообщение об изделиях, производимых на предприятиях, и передает готовое сообщение актору «Менеджер по продажам».

«Разослать рекламное сообщение клиентам» – «Менеджер по продажам» берет рекламное сообщение, переданное «Специалистом рекламного отдела», и рассылает его определенным клиентам (в данной работе не акцентируется, каким клиентам и как именно).

«Составить приказ об обновлении функций сотрудника» – актер «Начальник отдела» назначает определенному сотруднику новые обязанности в связи с, например, повышением, или переводом на другую должность. Далее об этом приказе уведомляется администратор и обновляет список ролей (см. ниже).

«Обновить список пользователей» – актер «Администратор» добавляет каждого работника предприятия в базу данных для работников. Для клиентов существует отдельная БД.

«Обновить список ролей и прав доступа» – «Администратор» может назначить на определенную роль пользователя (после уведомления начальника отдела). Например, просматривать отчеты могут только работники бухгалтерии, экономисты и начальники, а доступ к платежным документам есть только у юристов. «Администратор» обязан следить за актуальным списком ролей в каждом отделе.

«Настроить права пользователей» – «Администратор» может дать ряду сотрудников определенные права и назначить их на определенные роли.

«Обновить каталог изделий» включает в себя 2 ВИ «Обновить цены» и «Обновить список изделий». Обновлять каталог могут акторы: «Менеджер по продажам», «Экономист», «Специалист технического отдела», «Специалист рекламного отдела». Для этих акторов выполнена генерализация через одного условного актора «Специалист по наполнению каталога».

«Оформить базу клиентов» – вести БД со списком клиентов и их заказов могут акторы «Менеджер по продажам» и «Бухгалтерия». Выполнена генерализация, создан актер «Специалист по ведению базы клиентов».

«Принять заявку на заказ» может «Менеджер по производству». Заявка на изготовление определенных товаров может быть оформлена клиентом каким-либо образом через менеджера по продажам.

«Оформить ответ по заявке» включает в себя 2 ВИ, представляющие собой отрицательный ответ «Оформить невозможность выполнения заказа» и положительный «Подготовить данные для договора» (затем эти данные передаются актору «Юрист»). Этим действием занимается «Менеджер по продажам». Он анализирует оставленную клиентом заявку, и дает по ней ответ. Отрицательный ответ может быть дан в случае невыполнимой по каким-то причинам заявки. Если дан положительный ответ, то менеджер по продажам передает данные заявки/заказа актору «Юрист» для составления договора между фирмой и клиентом.

«Составить договор на выполнение заказа» – актер «Юрист» принимает заявку от «Менеджера по продажам» и составляет договор между фирмой и клиентом. Далее этот

договор передается обратно менеджеру по продажам.

«Выставить счет на оплату заказа и передать договор» – актер «Менеджер по продажам» после составления договора выставляет счет на оплату заказанного товара клиенту, и далее счет передается актору «Бухгалтерия».

«Составить спецификацию по заказу для производства» – «Менеджер по продажам» берет из заявки клиента данные по изделиям, которые компания должна изготовить для клиента. Он составляет спецификацию по изделиям и передает ее актору «Менеджер по производству», на производство, для изготовления товаров.

«Выполнить заказ в производстве и передать на отгрузку» – после выполнения заказа на производстве, «Менеджер по производству» уведомляет об этом актора «Менеджера по продажам» и передает заказ на отгрузку, и далее отгрузкой заказа занимается «Менеджер по продажам».

«Получить оплату» – актер «Менеджер по продажам» может получить оплату каким-либо способом. В качестве расширения указаны способы получения оплаты: ВИ «Получить платеж на банковский счет» и ВИ «Получить гарантийное письмо об оплате в течение месяца» (это – реальный и довольно распространенный способ оплаты, “рассрочка” на месяц или более, когда предварительно в компанию высылается гарантийное письмо). Но возможны и другие способы оплаты, в данной диаграмме не указанные, например, платеж через кассу.

«Оформить отгрузку заказа» – актер «Менеджер по продажам» распоряжается о доставке заказа клиенту.

4.2. Диаграмма классов

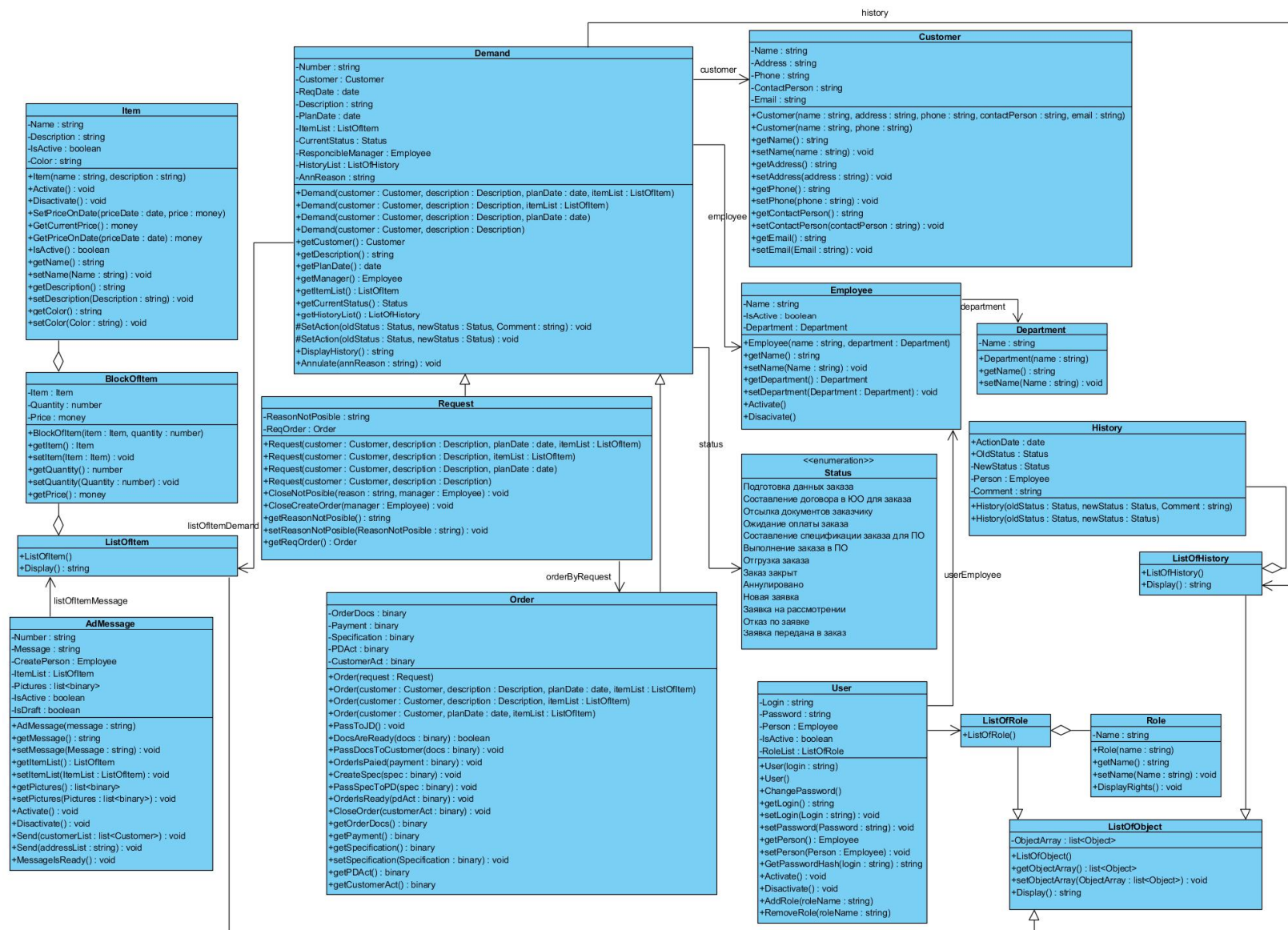


Рисунок 2 - Диаграмма классов (все классы)

На рисунке 2 классы показаны довольно мелко, детализация для классов дана ниже на рисунках 3, 4, 5, 6.

Классы Item, Demand, BlockOfItem, ListOfItem, Request:

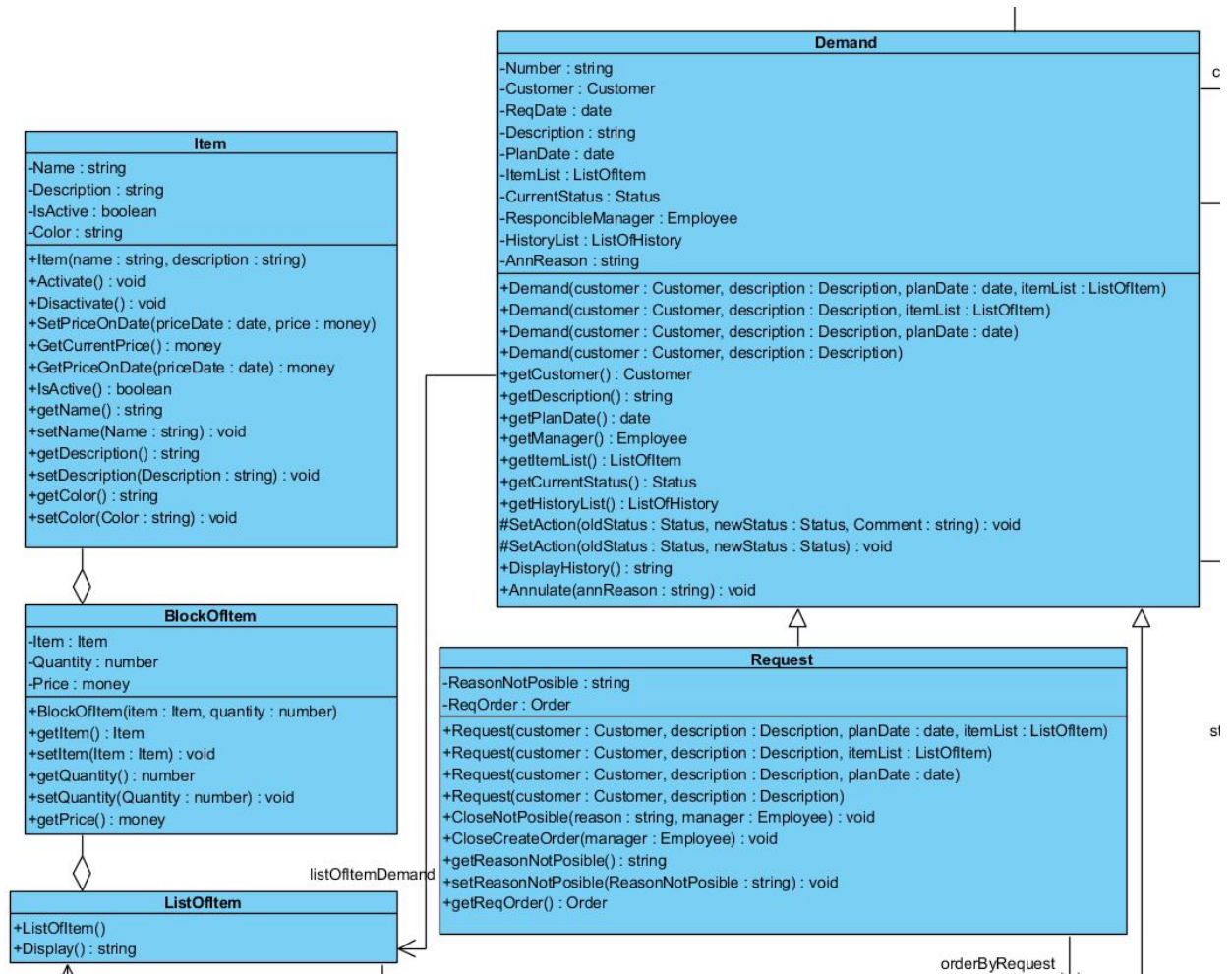


Рисунок 3 - Классы Item, Demand, BlockOfItem, ListOfItem, Request

Классы AdMessage и Order:

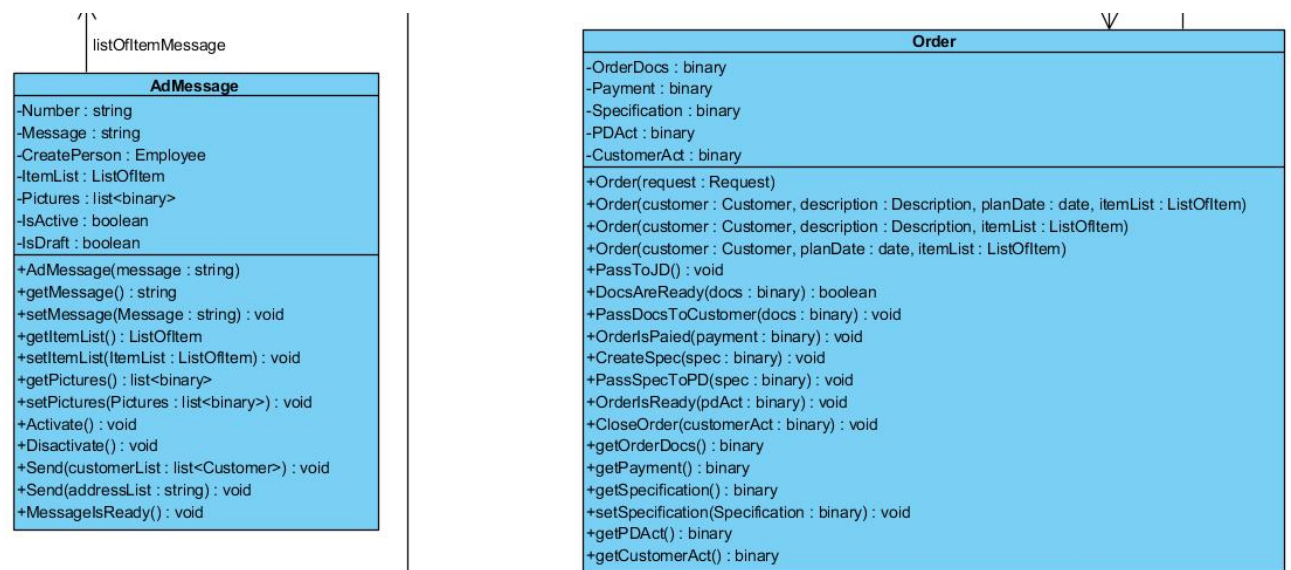


Рисунок 4 - Классы AdMessage и Order

Классы Customer, Employee, Department, History, ListOfHistory, Status:

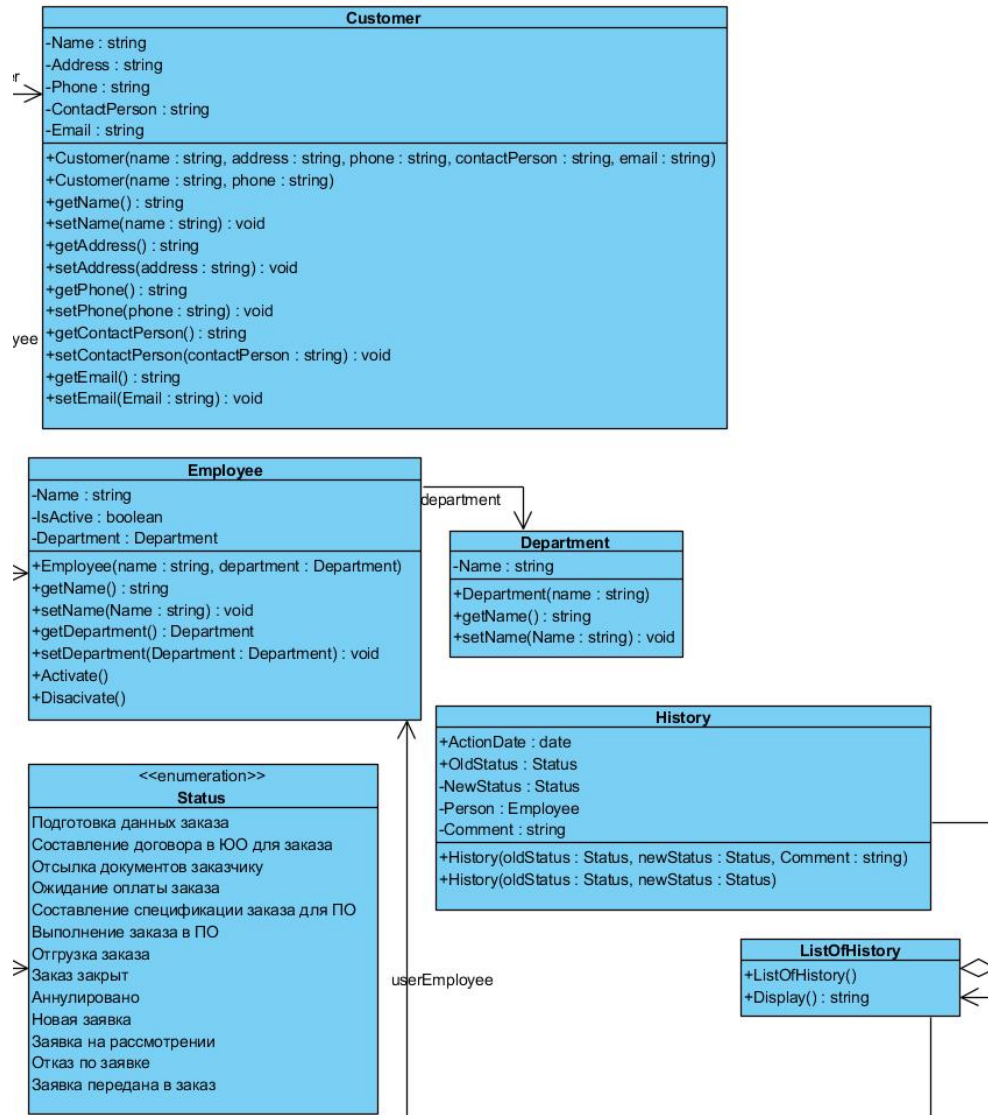


Рисунок 5 - Классы Customer, Employee, Department, History, ListOfHistory, Status

Классы User, Role, ListOfRole, LostOfObject:

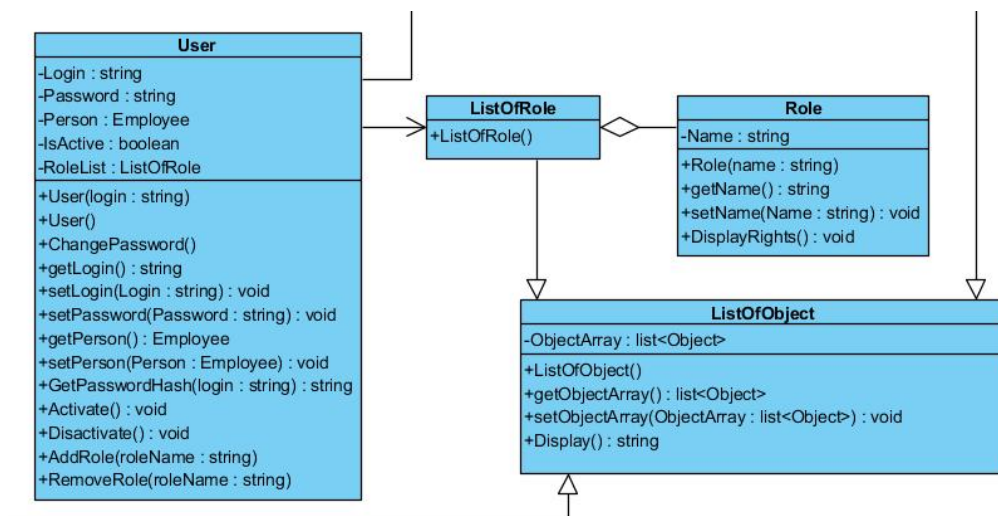


Рисунок 6 - Классы User, Role, ListOfRole, LostOfObject

4.2.1. Описание диаграммы классов

4.2.1.1. Описание класса «ListObject»

Класс «ListObject» – следует начать рассмотрение классов с этого, т.к. на диаграмме классов показаны 3 класса-списка (массива), содержащие список — несколько объектов других классов. Класс «ListObject» – служебный класс-родитель, он отображает общие методы и атрибуты классов-списков. Содержит метод «Display», позволяющий вывести на экран объект-список целиком (например, список товаров для заказа), и стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.2. Описание класса «Item»

Класс «Item» создан для описания объекта «Товар». Товар может быть активным (его можно заказать) или неактивным (временно снят с производства) – это отображает атрибут isActive. Также имеет атрибуты – название, описание, цвет. Товар можно активировать (метод Activate() : void) или сделать его неактивным (метод Disactivate() : void). Класс имеет метод GetPriceOnDate(priceDate : date) : money – будет взята цена данного товара для указанной даты из журнала «Товар – Цена – Дата» (в данной КР не рассматривается), и SetPriceOnDate (priceDate : date, price : money) – будет установлена указанная цена для указанной даты в этом журнале. Остальные методы – стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.3. Описание класса «BlockOfItem»

Класс «BlockOfItem» создан для описания объекта «1 товар в заказе» – строка в списке заказанных товаров, отображающая сам товар, количество в заказе и цену на сегодняшний день (для указанного количества).

4.2.1.4. Описание класса «ListOfItem»

Класс «ListOfItem» создан для описания всего списка заказанных товаров. Является потомком класса «ListObject». Объект класса содержит несколько объектов класса «BlockOfItem». Методы – стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.5. Описание класса «AdMessage»

Класс «AdMessage» применим для объектов – рекламных сообщений. Имеет атрибуты – Message (текст сообщения), CreatePerson (сотрудник, создавший рекламу), ItemList типа ListOfItem (класс описан выше – список товаров, к которым применяется реклама, цена в данном случае не указывается), Pictures (рекламная картинка), IsActive (активна ли реклама) и IsDraft (является ли черновиком в процессе создания). Класс имеет методы Activate() и

Deactivate(), меняющие IsActive. Также класс имеет метод MessageIsReady(), который переводит рекламное сообщение из черновика в готовое для запуска рекламы (метод меняет IsDraft). Метод Send с 2-мя наборами параметров отправляет сообщение списку клиентов на ряд адресов. Остальные методы – стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.6. Описание класса «Demand»

Класс «Demand» – это класс-родитель для классов «Request» (заявка) и «Order» (заказ). Это два класса имеют много общего, заявка (объект типа «Request») может перерасти в заказ (объект типа «Order»). Имеет атрибуты: Customer (клиент), ReqDate (дата создания заявки или заказа), Description (описание), PlanDate (ориентировочная дата завершения заказа), ItemList типа ListOfItem (список товаров для заказа – см. класс «ListOfItem»), CurrentStatus (статус заявки/заказа), ResponsibleManager (ответственный менеджер по продажам, ведущий заказ/заявку), HistoryList типа «ListOfHistory» (см. класс «ListOfHistory» - показывает историю заказа/заявки, отображает каждое изменение статуса), AnnReason (причина отмены, если заказ или заявка были отменены). Помимо конструктора, getter'ов и setter'ов имеет protected метод SetAction с 2-мя вариантами вызова: #SetAction(oldStatus : Status, newStatus : Status, Comment : string) и #SetAction(oldStatus : Status, newStatus : Status), меняющим текущий статус и сохраняющим старый и новый статус в объект типа «ListOfHistory». Имеет метод DisplayHistory(), позволяющий смотреть историю заказа/заявки. Также имеет метод Annulate(annReason : string), который отменяет заказ или заявку с переданной причиной отмены.

4.2.1.7. Описание класса «Request»

Класс «Request» – потомок класса «Demand», отображает данные о заявке, которую клиент оставляет до оформления заказа, чтобы возможность ее исполнения была проверена компанией. Имеет атрибуты ReasonNotPossible (причина, по которой невозможно выполнить будущий заказ) и ReqOrder типа «Order» (оформленный заказ, если его выполнить можно). Класс имеет метод CloseNotPossible(reason : string, manager : Employee), который закрывает заявку по причине невозможности ее выполнения, с указанием причины отмены и менеджера, который вел заявку. Класс также имеет метод CloseCreateOrder(manager : Employee), который закрывает заявку и открывает заказ по ней, с передачей имени сотрудника. Остальные методы – стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.8. Описание класса «Order»

Класс «Order» – потомок класса «Demand», отображает данные о заказе, открытом после рассмотрения заявки. Атрибуты: OrderDocs (изображение – подтверждение заказа),

Payment (изображение – подтверждение оплаты заказа), Specification (изображение – составленная спецификация для изготовления заказа), PDAct (изображение – акт, подтверждающий полноту комплектации заказа после изготовления товаров), CustomerAct (изображение – подтверждение получения товара клиентом). Имеет методы: PassToJD (передать заказ юристу для составления контракта), DocsAreReady (подтвердить что контракт составлен с приложением контракта), PassDocToCustomer (передать полученные от юриста документы клиенту), OrderIsPaid (подтвердить факт оплаты от клиента), CreateSpec (создать спецификацию товаров для производства), PassSpecToPD (передать спецификацию отделу производства), OrderIsReady (подтвердить готовность заказа), CloseOrder (закрыть заказ). Остальные методы – стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.9. Описание класса «Status»

Класс «Status» – класс-перечисление (enumeration) возможных статусов атрибута Status класса «Demand».

4.2.1.10. Описание класса «Customer»

Класс «Customer» описывает информацию о покупателе товара. Имеет атрибуты – личные данные клиента. Методы класса – стандартные конструктор, setter'ы и getter'ы.

4.2.1.11. Описание класса «Employee»

Класс «Employee» описывает информацию о сотрудниках. Имеет атрибуты: Name (имя), Department (отдел) и IsActive (работает ли сотрудник, или находится в отпуске/на больничном/уволился). Сотрудник не удаляется из БД, чтобы не нарушать существующие связи в таблицах. Имеет методы Activate() и Disactivate(), меняющие статус сотрудника на активный или неактивный соответственно. Реализация в приложении/на сайте обычно производится добавлением 1 кнопки – “Activate” , если сотрудник неактивен, или “Disactivate”, если активен. Остальные методы – стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.12. Описание класса «Department»

Класс «Department» описывает отдел, которому принадлежит сотрудник. Содержит 1 атрибут Name и стандартные методы – конструктор класса, getter и setter. Вынесен в отдельный класс, т.к. сотрудник может принадлежать только существующему отделу, а не любому указанному, т.е. сделать атрибут Department в классе Employee типа string нельзя. Но список отделов может меняться, поэтому не используется класс-перечисление.

4.2.1.13. Описание класса «User»

Класс «User» описывает информацию о сотрудниках как о юзерах в базе данных сотрудников, имеющих определенные роли и доступ в личный кабинет. Имеются атрибуты Login, Password, Person типа Employee (существующий сотрудник), IsActive и RoleList типа «ListOfRoles» (см. описание ниже), показывающий список ролей для каждого юзера. Пример роли: просмотрщик отчетов, или наполняющий каталог товаров. Класс имеет методы: ChangePassword(), для смены пароля (для упрощения предполагается, что пароль назначается автоматически), Activate(), Deactivate(), AddRole(roleName) для присвоения юзеру определенной роли, DeleteRole(roleName) для удаления юзера из роли, и GetPasswordHash(login). О методе GetPasswordHash следует рассказать отдельно. Во многих БД пароли юзеров хранятся не в прямом виде, а в зашифрованном (например, методом шифрования base64). Авторизация юзера происходит так: введенный на сайте/в приложении пароль зашифровывается, из БД также достается зашифрованный пароль, и значения паролей сравниваются. Метод GetPasswordHash как раз достает из базы данных значение пароля в зашифрованном виде. Входной параметр метода – login юзера. Другие методы класса – стандартные конструктор класса, getter'ы и setter'ы. Класс не совмещен с классом «User», т.к. не каждый работник может иметь личный кабинет и быть назначенным на роли.

4.2.1.14. Описание класса «Role»

Класс «Role» описывает одну роль, на которую может быть назначен юзер. С помощью метода DisplayRights() отображаются права, доступные юзеру, назначенному на данную роль. Остальные методы – стандартные конструктор класса, getter'ы и setter'ы.

4.2.1.15. Описание класса «ListOfRole»

Класс «ListOfRole» содержит информацию о списке ролей. 1 юзер может быть назначен сразу на несколько ролей – например, один человек может и составлять отчеты по плану продаж, и наполнять каталог товаров. Это потомок класса «ListOfObject», наследующий все его атрибуты и методы.

4.2.1.16. Описание класса «History»

Класс «History» содержит информацию об 1 строке в истории заказа. Клиент в любой момент может посмотреть всю историю заказов: в какой момент заявка была одобрена и был создан заказ, в какой момент прошла оплата, в какой момент заказ был изготовлен и т.д. Класс содержит атрибуты: ActionDate (дата изменения статуса), OldStatus (предыдущий статус), NewStatus (текущий статус), Person типа «Employee» (сотрудник, сменивший статус заказа), Comment (комментарий к новому статусу). Класс содержит только метод-

конструктор, т.к. ни один атрибут не может быть изменен, и просмотр доступен для всего объекта целиком.

4.2.1.17. Описание класса «ListOfHistory»

Класс «ListOfHistory» содержит всю историю заказа – все изменения статусов заявки/заказа, с момента создания. Это потомок класса «ListOfObject», наследующий все его атрибуты и методы. Содержит только конструктор и метод Display(), отображающий всю историю целиком. В качестве примера-ориентира (не конкретно объекта этого класса, а похожего из реальной жизни) можно привести отображение статусов посылки на сайте «Почты России». После ввода номера посылки можно увидеть список статусов – «Заказ в сортировочном центре», «Заказ прибыл в ваш почтовый офис», «Заказ доставлен»; дату выставления каждого статуса, почтовый индекс, соответствующий местонахождению посылки в данный момент, город и страну.

4.3. Диаграммы последовательности

Диаграмма последовательностей «Заявка и заказ товара»:

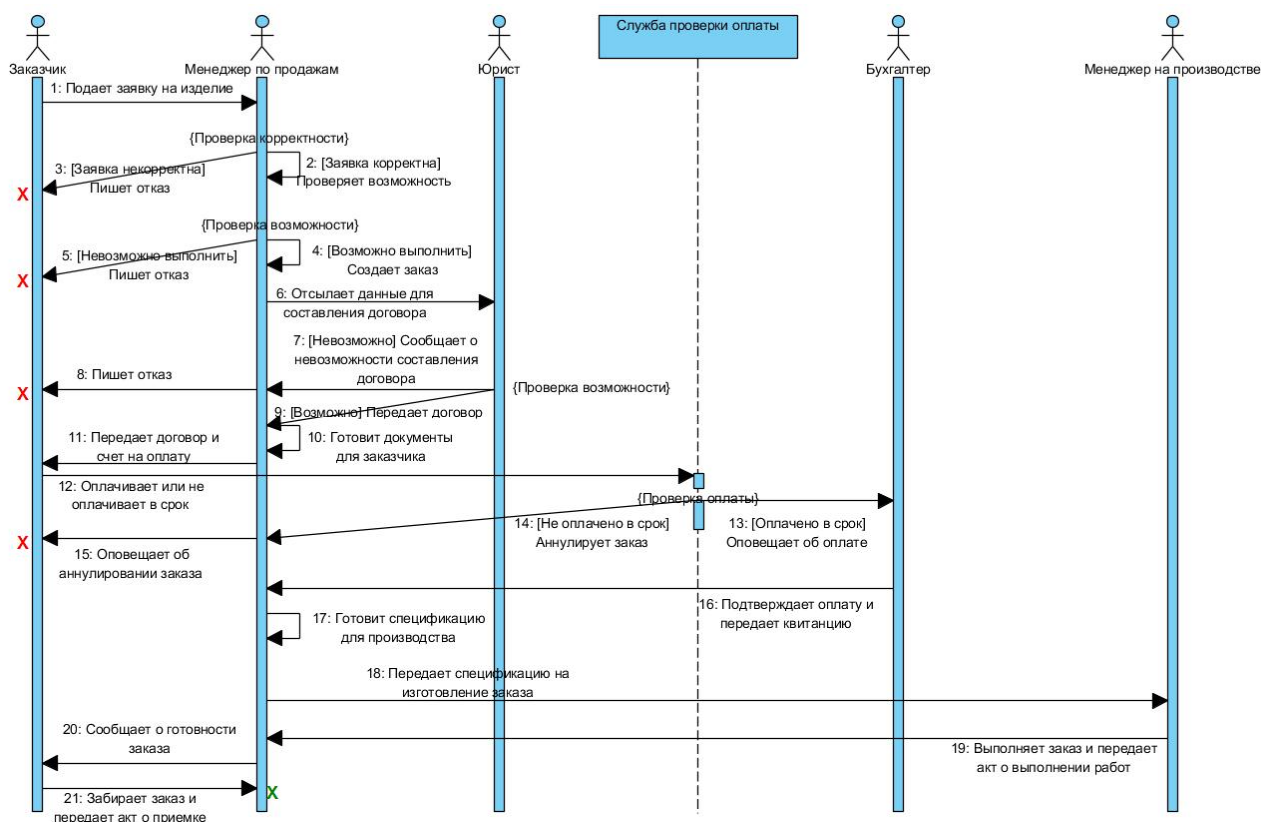


Рисунок 7 - Диаграмма последовательностей «Заявка и заказ товара»

Диаграмма последовательностей «Создание рекламного сообщения»:

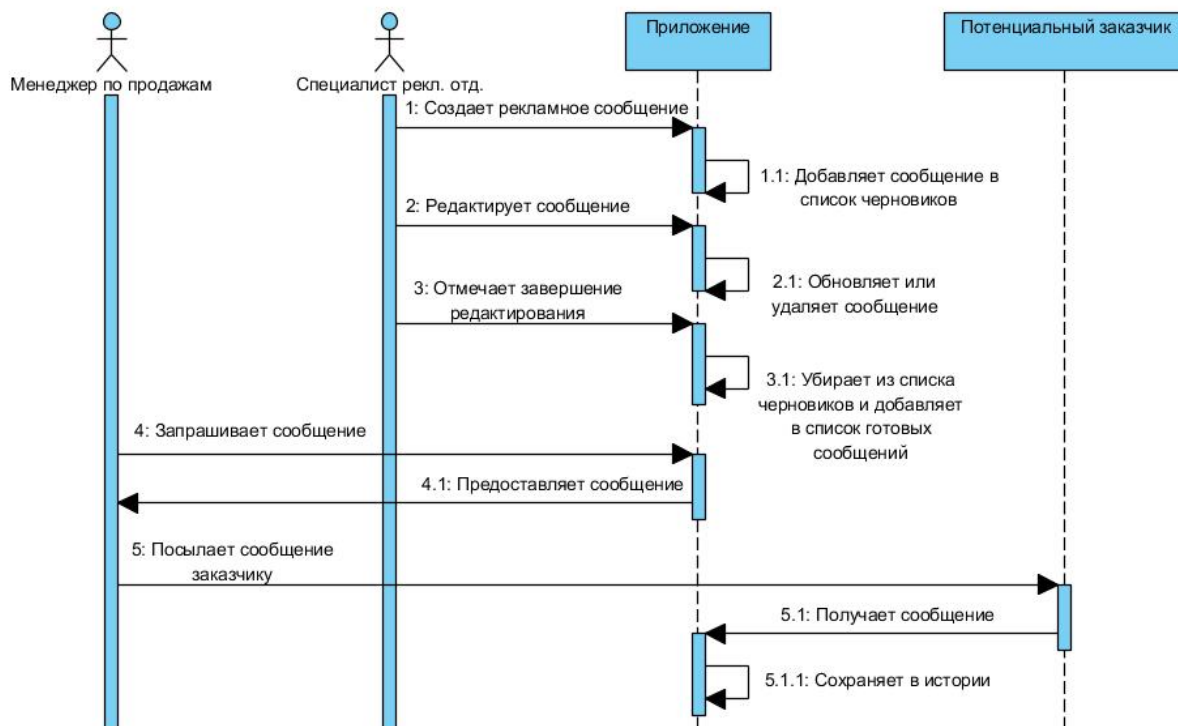


Рисунок 8 - Диаграмма последовательностей «Создание рекламное сообщение»

Диаграмма последовательностей «Обновление каталога изделий»:

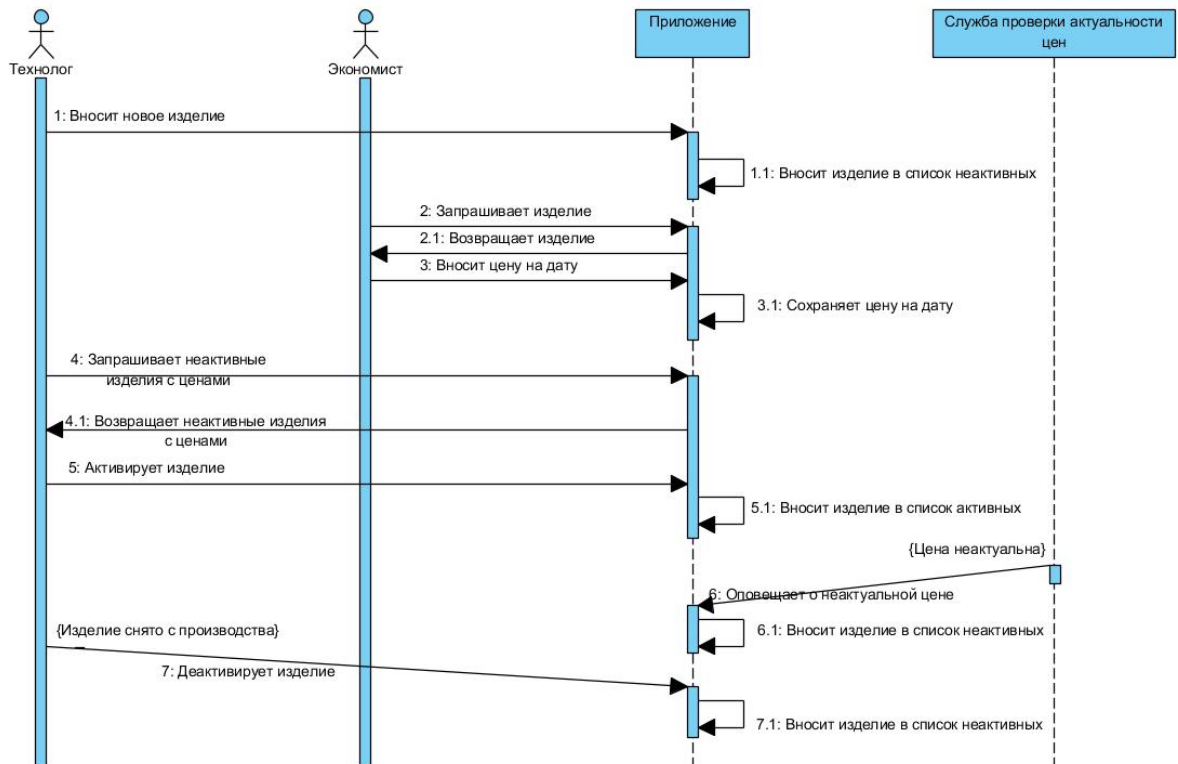


Рисунок 9 - Диаграмма последовательностей «Обновление каталога изделий»

Диаграмма последовательностей «Цикл пользователя»:

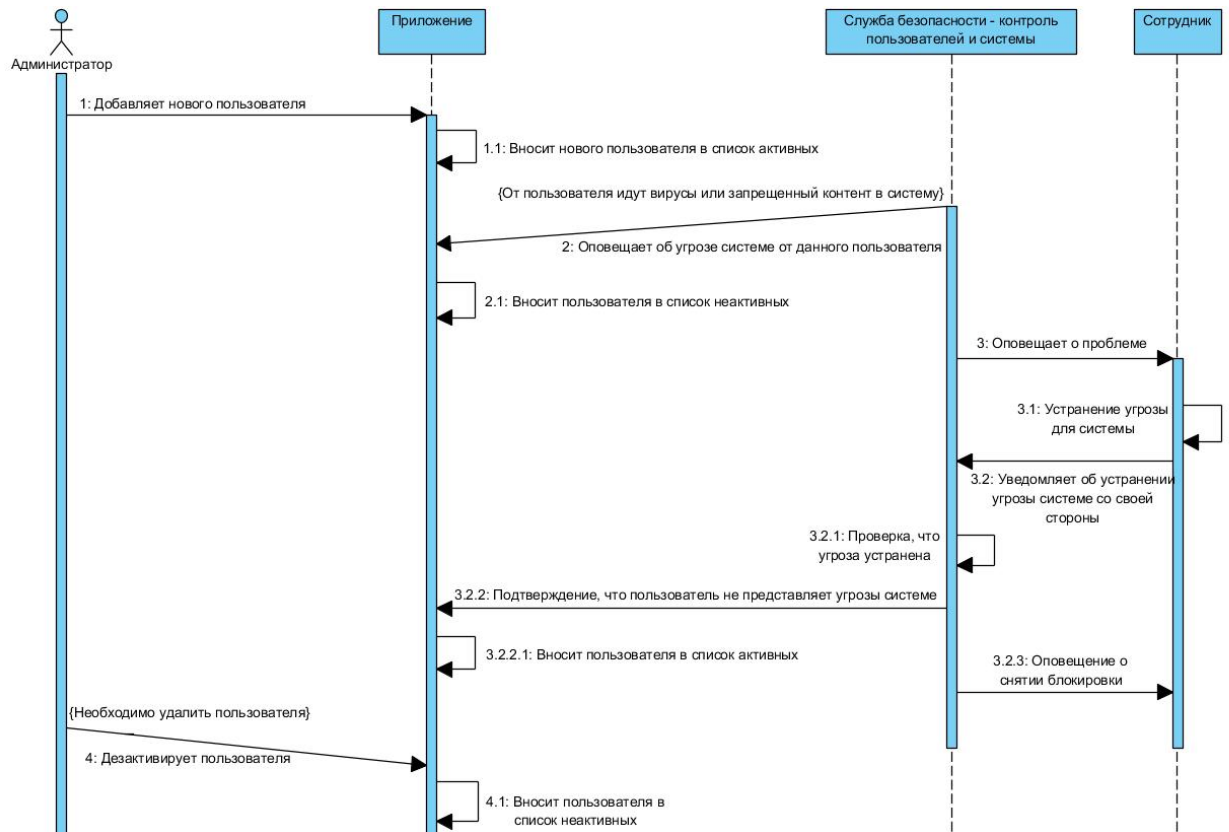


Рисунок 10 - Диаграмма последовательностей «Цикл пользователя»

Диаграмма последовательностей «Назначение и удаление пользователя на роль/из роли»:

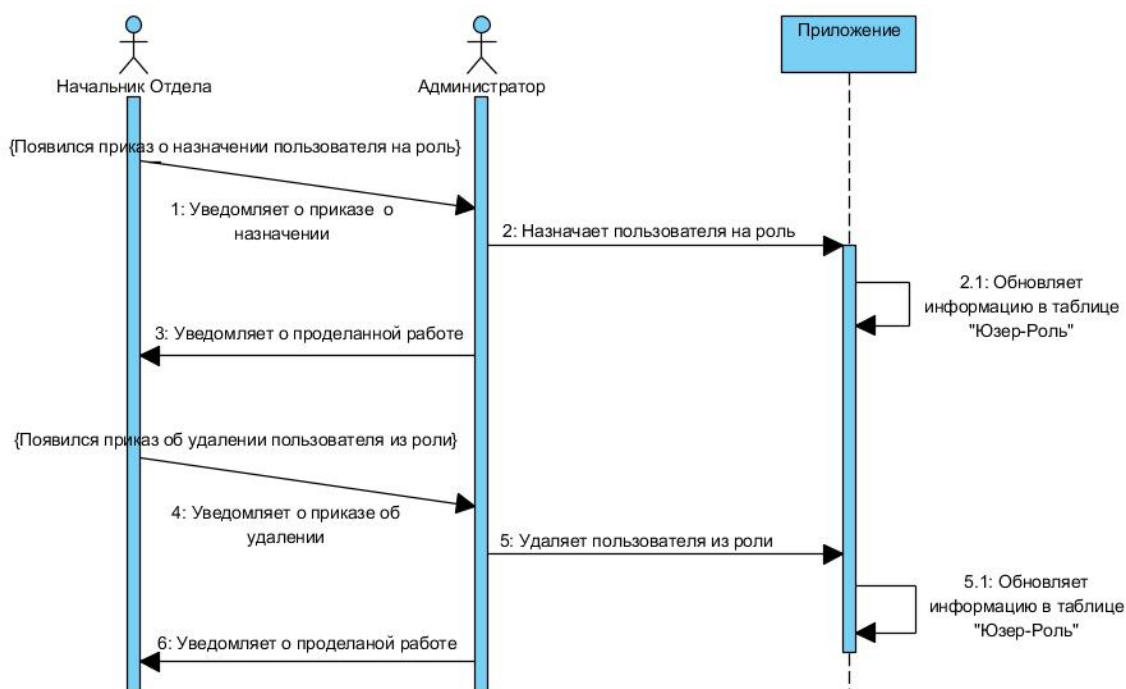


Рисунок 11 – Диаграмма последовательностей «Назначение и удаление пользователя на роль/из роли»

4.3.1. Описания диаграмм последовательности

4.3.1.1. Описание диаграммы «Заявка и заказ товара»

Данная диаграмма описывает последовательность действий и состояний заявки и открытому по данной заявке заказу.

Охватывает ВИ: «Принять заявку на заказ» + «Оформить ответ по заявке» + «Оформить невозможность вып. заказа» + «Подготовить данные для договора» + «Составить договор на выполнение заказа» + «Выставить счет на оплату заказа и передать договор» + «Составить спецификацию по заказу для производства» + «Выполнить заказ в производстве и передать на отгрузку» + «Получить оплату» + «Получить платеж на банковский счет» + «Получить гарант. письмо об оплате в теч. месяца» + «Оформить отгрузку заказа».

1: Заказчик подает новую заявку, она попадает менеджеру по продажам. Идет проверка корректности.

2: Если заявка корректна, менеджер по продажам начинает первично проверять возможность изготовления товара, чтобы отсеять заведомо невыполнимые заявки. Идет проверка возможности выполнения будущего заказа.

3: Если заявка некорректна, менеджер по продажам пишет отказ и отправляет отказ заказчику. Завершение сценария.

4: Если заказ выполнить реально, менеджер по продажам создает заказ.

5: Если заказ выполнить нереально, менеджер по продажам пишет отказ и отправляет

отказ заказчику. Завершение сценария.

6: Менеджер по продажам открывает новый заказ и отправляет данные для составления договора между заказчиком и фирмой-изготовителем юридическому отделу. Юрист проверяет возможность создания договора.

7: Если юридический отдел постановил, что договор составить невозможно, менеджер по продажам получает уведомление об этом.

8: Менеджер по продажам пишет отказ и отправляет отказ заказчику. Завершение сценария.

9: Если договор составить можно, юридический отдел составил договор. Юрист передает договор менеджеру по продажам.

10: Менеджер по продажам готовит документы для передачи заказчику.

11: Менеджер по продажам передает документы и счет на оплату заказчику. Идет ожидание оплаты.

12: Заказчик оплачивает договор, или не оплачивает его и выходит время, отведенное на оплату. Данные (уведомление или то, что истек срок для оплаты заказа) автоматически передаются от клиента в службу проверки оплаты. Идет проверка факта оплаты.

13: Если оплата поступила, служба проверки оплаты автоматически оповещает бухгалтерию о факте оплаты заказа.

14: Если оплата не поступила, служба проверки оплаты автоматически аннулирует заказ и уведомляет об этом менеджера по продажам.

15: Менеджер по продажам оповещает об аннулировании заказа клиента.

16: Бухгалтер подтверждает факт оплаты, менеджер по продажам получает квитанцию об оплате из бухгалтерии.

17: Менеджер по продажам готовит спецификацию на заказанные товары для отдела производства. В спецификацию входят: описание комплектации товаров, цвет, размер, и т.д.

18: Менеджер по продажам передает спецификацию на заказ менеджеру по производству.

19: Отдел производства выполнил заказ. Менеджер по производству отправляет уведомление о готовности заказа менеджеру по продажам.

20: Менеджер по продажам посылает уведомление заказчику о готовности заказа

21: Заказчик забирает заказ или получает отгруженный заказ, подписывает акт о приеме заказа и отправляет этот акт менеджеру по производству. Завершение сценария.

4.3.1.2. Описание диаграммы «Создание рекламного сообщения»

Данная диаграмма описывает последовательность действий при создании нового

рекламного сообщения.

Охватывает ВИ: «Составить рекламное сообщение» + «Разослать рекламное сообщение клиентам».

1: Специалист рекламного отдела создает черновик (макет, заготовку) для будущего рекламного сообщения.

1.1: Приложение, в котором создавался черновик сообщения, сохраняет его и хранит в списке черновиков.

2: Специалист рекламного отдела редактирует в приложении сохраненный черновик, детализирует сообщение, проверяет на наличие ошибок; либо запускает удаление черновика, если он оказался неудачным или более неактуальным.

2.1: Рекламное сообщение отредактировано или удалено в приложении.

3: Специалист рекламного отдела завершает редактирование и отмечает в приложении, что сообщение готово для перевода в категорию «активные».

3.1: Приложение удаляет готовое сообщение из раздела «черновики» и добавляет в категорию «активные». Активные рекламные сообщения доступны для использования и рассылке клиентам, специалист рекламного отдела гарантирует актуальность рекламных сообщений в категории «активные».

4: Менеджер по продажам запрашивает в приложении готовое рекламное сообщение для рассылки его потенциальным заказчикам.

4.1: Менеджер по продажам получает из приложения готовое рекламное сообщение.

5: Менеджер по продажам отправляет рекламное сообщение потенциальному заказчику, которого может заинтересовать категория товаров, которую рекламирует данное сообщение.

5.1: Потенциальный заказчик получает рекламное сообщение, и этот факт регистрируется в приложении.

5.1.1: Факт получения потенциальным заказчиком рекламного сообщения фиксируется в приложении, чтобы не допустить будущего «спама» этого сообщением для клиента.

4.3.1.3. Описание диаграммы «Обновление каталога изделий»

Данная диаграмма описывает последовательность действий и состояний товара.

Охватывает ВИ: «Обновить каталог изделий» + «Обновить цены» + «Обновить список изделий» + «Установить цены».

1: Специалист технологического отдела разрабатывает новое изделие и вносит его в электронный каталог – приложение.

1.1. По умолчанию в приложении изделие помещается в список неактивных.

- 2: Экономист запрашивает в приложении информацию о новом изделии.
- 2.1: Экономист получает данные о новом изделии из приложения.
- 3: Экономист устанавливает цену на изделие согласно журналу с ценами по датам.
- 3.1: Приложение сохраняет установленную цену на изделие.
- 4: Технолог посылает запрос в приложение для данного изделия
- 4.1: Приложение возвращает информацию о неактивных изделиях
- 5: Технолог помечает изделие как активное и отправляет информацию об этом в приложение
- 5.1: Приложение вносит изделие в список активных. Изделие доступно для заказа.
- 6: Когда цена перестает соответствовать журналу цен на товары по датам, т.е. становится неактуальной, служба проверки актуальности цен автоматически отправляет информацию об этом в приложение.
- 6.1: Приложение вносит изделие в список неактивных. Товар недоступен к заказу.
- 7: Когда изделие снимается с производства, технолог деактивирует изделие и отправляет информацию об этом в приложение.
- 7.1: Приложение вносит изделие в список неактивных. Товар недоступен к заказу.

4.3.1.4. Описание диаграммы «Цикл пользователя»

Данная диаграмма описывает последовательность для оформления нового пользователя в системе компании и возможные ветви сценария, связанные с деятельностью пользователя. Созданный аккаунт пользователя привязывается к сотруднику.

Охватывает ВИ: «Обновить список пользователей», «Настроить права пользователей».

В данном разделе реализована ветвь сценария, при которой активный пользователь, обладающий доступом в систему и определенными правами, поставил нелицензионный софт, содержащий вирусы, на свой компьютер, и вирусы начали проникать в систему. К сожалению, такая версия сценария встречается в реальной жизни не так редко.

1: Администратор добавляет нового пользователя в систему и посылает запрос на добавление в приложение.

1.1: Приложение вносит нового пользователя в список активных, т.е. активным становится сразу любой новый пользователь. Активный пользователь может входить в систему предприятия по логину и паролю, искать определенную информацию и т.д.

2: Если от пользователя в сеть предприятия начинают лезть вирусы, служба безопасности (СБ) обнаруживает это и посылает в приложение запрос на временную блокировку пользователя для защиты системы.

2.1: Приложение автоматически делает пользователя неактивным (блокирует). Пользователь не может входить по логину. Уведомление администратора

3: СБ шлет уведомление на email сотрудника, связанного с пользователем, о проблемах с его аккаунтом и требует их устранить – удалить все нелицензионное ПО и запрещенный контент со своего компьютера, а также проверить все диски компьютера антивирусом.

3.1: Сотрудник выполняет требование СБ.

3.2: Сотрудник уведомляет СБ о том, что угроза системе с его стороны устранена.

3.2.1: СБ проверяет, что угроза действительно устранена.

3.2.2: Когда проблема устранена и угрозы системе со стороны пользователя нет, и снимают блокировку в приложении для данного пользователя.

3.2.2.1: Приложение вносит пользователя в список активных.

4: Если необходимо по каким-либо причинам деактивировать пользователя, администратор деактивирует пользователя в приложении. Возможные причины деактивации: увольнение сотрудника или создание нового аккаунта-пользователя для сотрудника. Отпуск или больничный не являются причиной деактивации пользователя.

4.1: Приложение вносит пользователя в список неактивных.

4.3.1.5. Описание диаграммы «Назначение и удаление пользователя на роль/из роли»

Данная диаграмма описывает последовательность при назначении пользователя на какую-либо роль либо удаление его из этой роли. Пример ролей: специалист по наполнению базы данных клиентов, просмотрщик отчетов, специалист по наполнению каталога товаров.

Охватывает ВИ: «Обновить список ролей и прав доступа», «Составить приказ об обновлении функций сотрудника».

1: Начальник отдела составляет приказ о назначении определенного пользователя на определенную роль и уведомляет об этом администратора.

2: Администратор шлет запрос в приложение о назначении пользователя на роль (обычно это означает добавление строки в БД таблице «Юзер – Роль»).

2.1: Приложение выполняет добавление информации о связи пользователя с ролью.

3: Администратор уведомляет начальника отдела о назначении пользователя на роль.

4: Начальник отдела составляет приказ об удалении определенного пользователя из определенной роли и уведомляет об этом администратора.

5: Администратор шлет запрос в приложение об удалении пользователя из роли (обычно это означает удаление строки в БД таблице «Юзер – Роль»).

- 5.1: Приложение выполняет добавление информации о связи пользователя с ролью.
- 6: Администратор уведомляет начальника отдела об удалении пользователя из роли.

4.4. Диаграммы состояний

Диаграмма состояний заявки (класс «Request»):

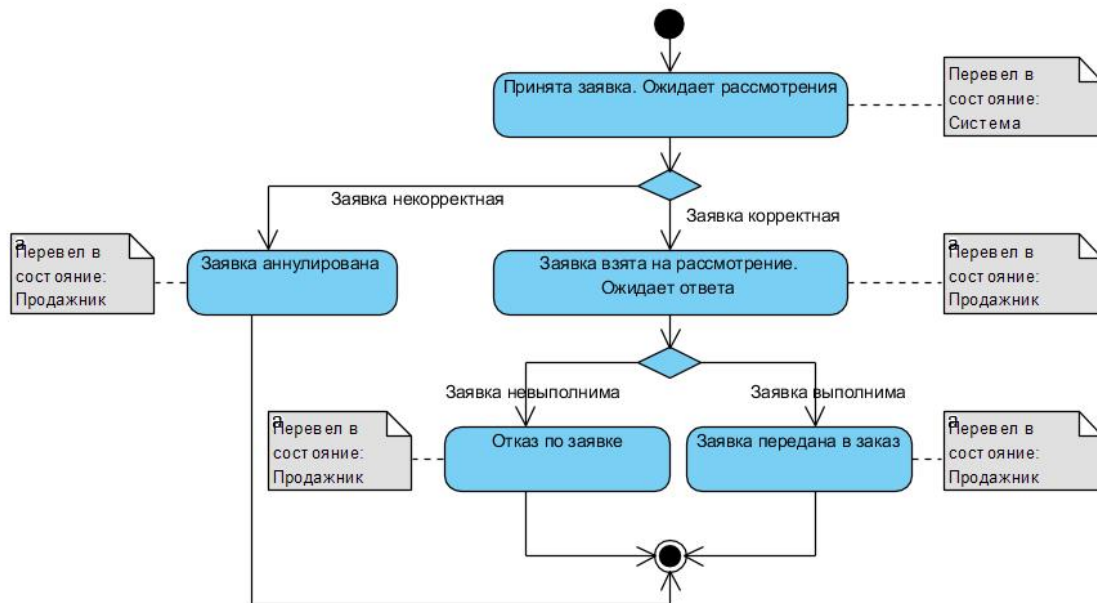


Рисунок 12 - Диаграмма состояний заявки (класс «Request»)

Диаграмма состояний заказа (класс «Order»):

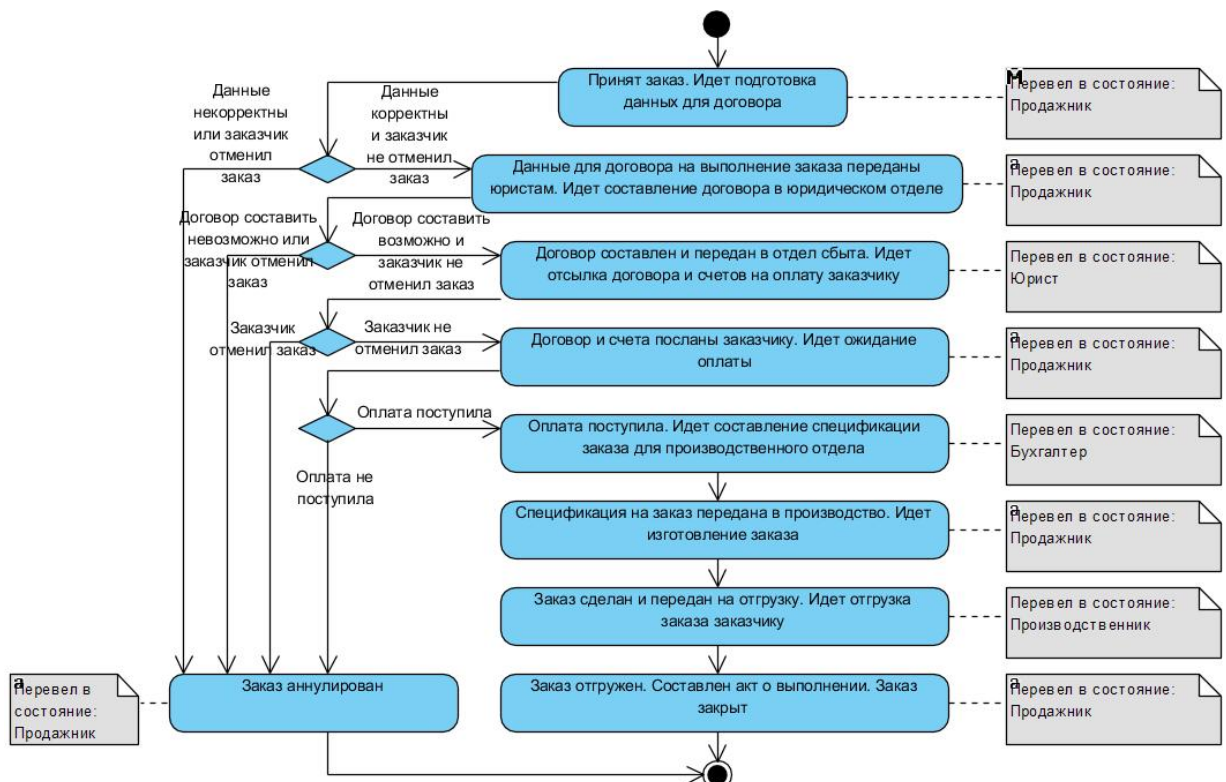


Рисунок 13 - Диаграмма состояний заказа (класс «Order»)

Диаграмма состояний сотрудника (класс «Employee»):

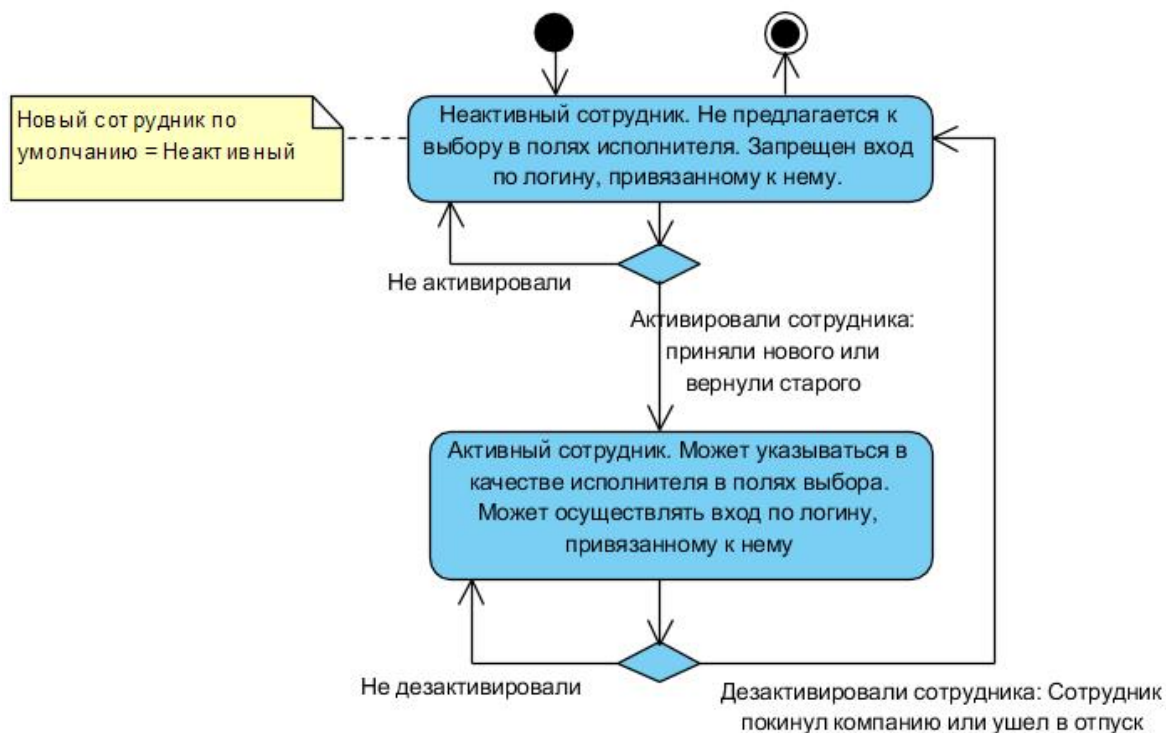


Рисунок 14 - Диаграмма состояний сотрудника (класс «Employee»)

Диаграмма состояний пользователя (класс «User»):

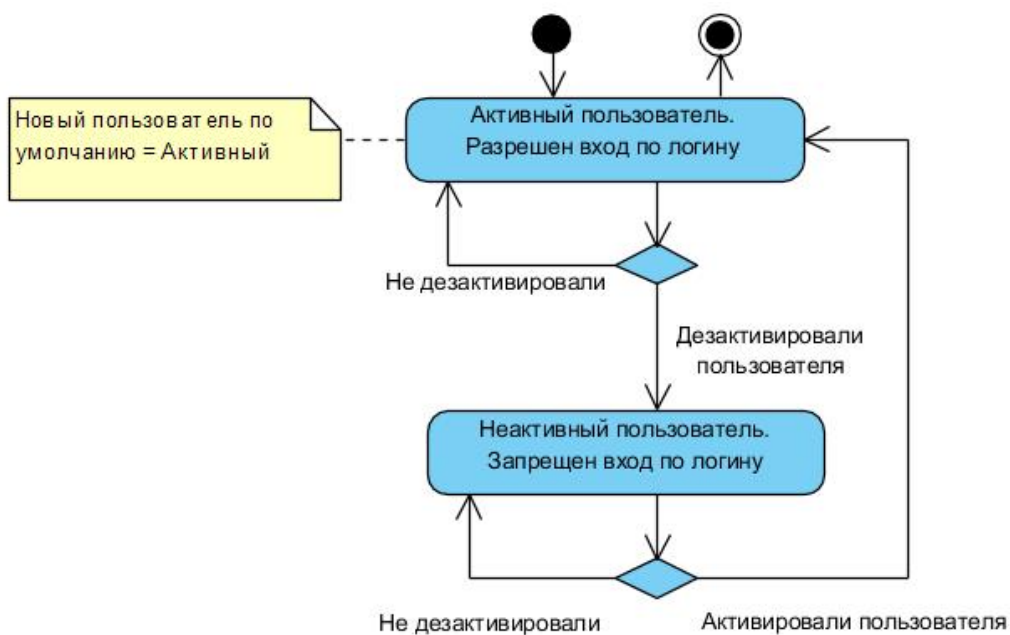


Рисунок 15 - Диаграмма состояний пользователя (класс «User»)

Диаграмма состояний рекламного сообщения (класс «AdMessage»):

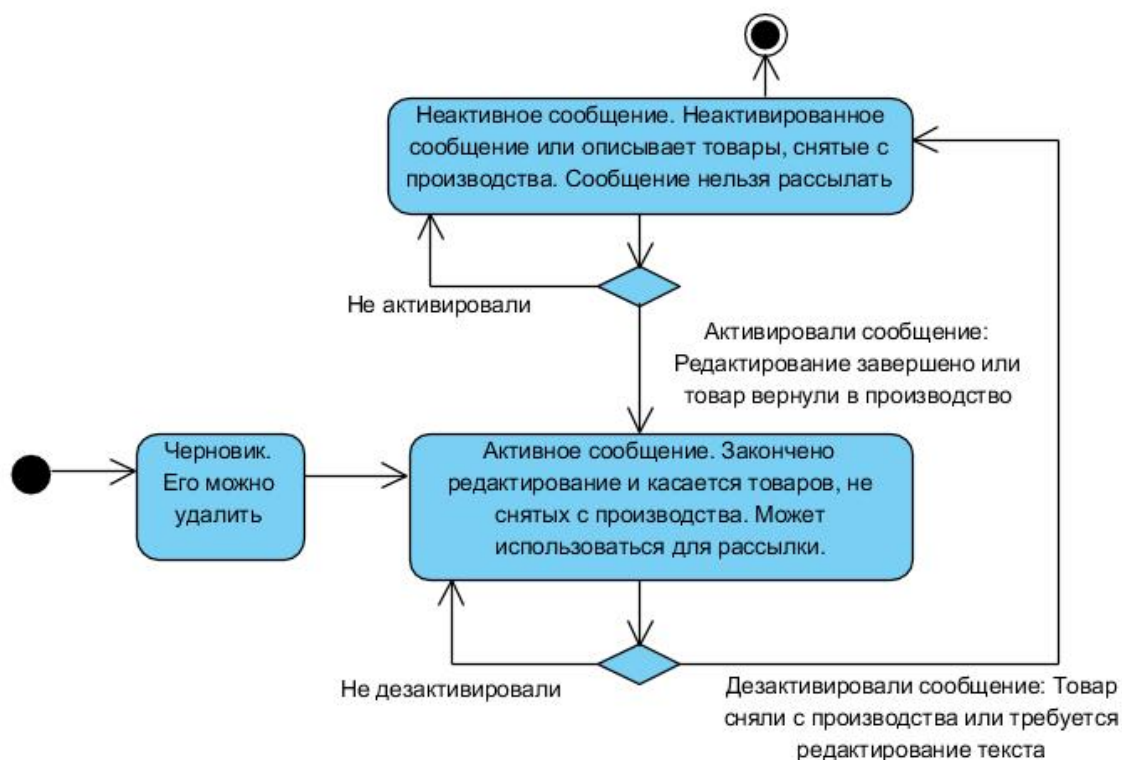


Рисунок 16 - Диаграмма состояний рекламного сообщения (класс «AdMessage»)

Диаграмма состояний товара (класс «Item»):

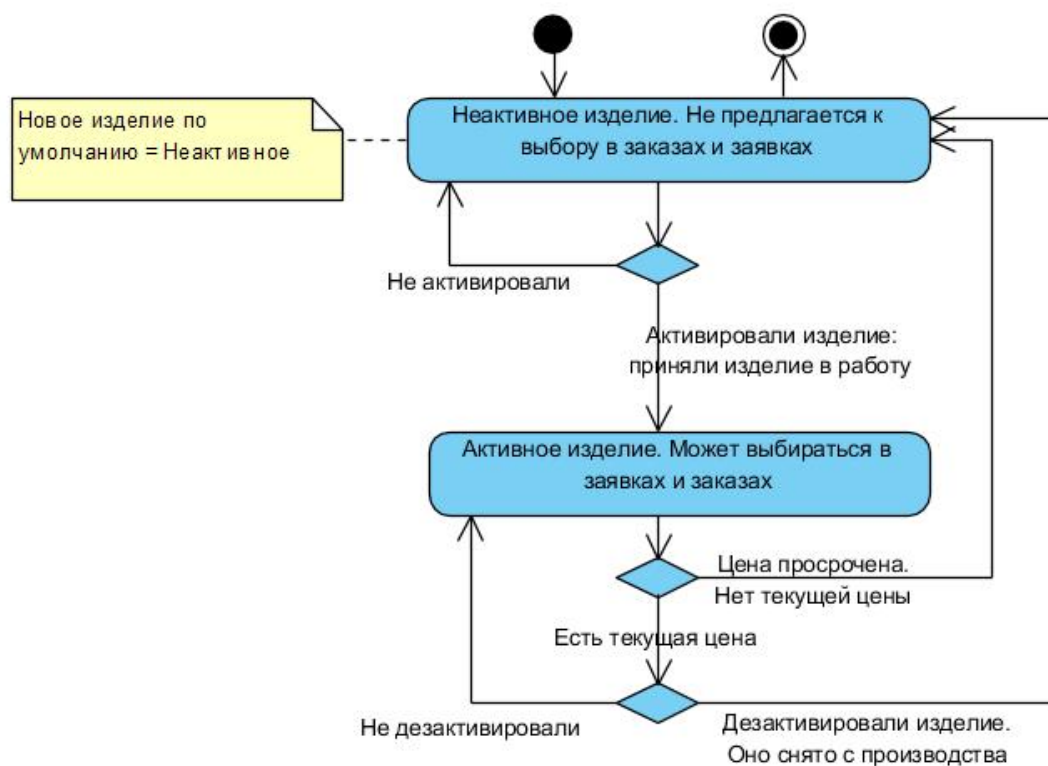


Рисунок 17 - Диаграмма состояний товара (класс «Item»)

4.4.1. Описание диаграмм состояний

4.4.1.1. Описание состояний объекта класса «Request» (заявка)

Состояние 1 «Принята заявка. Ожидает рассмотрения»: клиент зарегистрировал новую заявку на изготовление товаров. Заявка находится в очереди на рассмотрение. Переход в состояние 2, если заявка корректна, или в состояние 3, если некорректна. Начальное состояние.

Состояние 2 «Заявка взята на рассмотрение. Ожидает ответа»: менеджер по продажам принял заявку на рассмотрение и изучает ее. Переход в состояние 4, если заявка выполнима, или в состояние 5, если невыполнима.

Состояние 3 «Заявка аннулирована»: менеджер по продажам первично изучил заявку и постановил, что она некорректная или заведомо невыполнимая. Конечное состояние.

Состояние 4 «Заявка передана в заказ»: менеджер по продажам более детально изучил заявку, проконсультировался с производственным отделом, получил подтверждение, что заявку выполнить представляется возможным. В результате формируется новый объект типа «Order» (Заказ). Конечное состояние.

Состояние 5 «Отказ по заявке»: менеджер по продажам более детально изучил заявку, проконсультировался с производственным отделом, получил уведомление о том, что заявка невыполнима на данный момент – на складе не хватает определенных деталей, либо какой-либо пункт прайс-листа успели убрать за время рассмотрения заявки, либо по другой производственной причине. Конечное состояние.

4.4.1.2. Описание состояний объекта класса «Order» (заказ)

Состояние 1 «Принят заказ. Идет подготовка данных для договора»: менеджер по продажам открыл заказ на товары после рассмотрения заявки и готовит данные для договора на оплату товара. Переход в состояние 2, если данные корректны и клиент не отменил заказ, или в состояние 9, если данные некорректны или клиент отменил заказ. Начальное состояние.

Состояние 2 «Данные для договора на выполнение заказа переданы юристам. Идет составление договора в юридическом отделе»: данные переданы юридическому отделу, и юристы составляют договор на оплату. Переход в состояние 3, если составить договор возможно и клиент не отменил заказ, или в состояние 9, если составление договора невозможно по каким-либо причинам или клиент отменил заказ.

Состояние 3 «Договор составлен и передан в отдел сбыта. Идет отсылка договора и счетов на оплату заказчику»: юридический отдел переслал готовый договор на оплату отделу сбыта, и менеджер по продажам находится в процессе отправки документов клиенту.

Переход в состояние 4, если клиент не отменил заказ, или в состояние 9, если клиент отменил заказ.

Состояние 4 «Договор и счета посланы заказчику. Идет ожидание оплаты»: менеджер по продажам отправил договор по заказу и счет на оплату заказа клиенту и находится в процессе ожидания оплаты. Переход в состояние 5, если оплата поступила, или в состояние 9, если клиент отменил заказ или время для оплаты товара вышло (установленный timeout).

Состояние 5 «Оплата поступила. Идет составление спецификации заказа для производственного отдела»: бухгалтерия подтвердила факт оплаты товара. Менеджер по продажам составляет спецификацию для заказанных товаров – комплектацию, цвет, размер изделий. Переход всегда в состояние 6. После поступления оплаты отмена заказа невозможна, т.к. в данной работе не рассматривается вариант возврата оплаты и возврата товара.

Состояние 6 «Спецификация на заказ передана в производство. Идет изготовление заказа»: менеджер по продажам закончил составлять спецификацию на заказ и передал ее менеджеру по производству. Отдел производства выполняет изготовление товаров заказа. Переход всегда в состояние 7.

Состояние 7 «Заказ сделан и передан на отгрузку. Идет отгрузка заказа заказчику»: все товары из заказа готовы, менеджер по производству подтвердил готовность. Менеджер по продажам контролирует погрузку и доставку готовых товаров клиенту. Переход всегда в состояние 8.

Состояние 8 «Заказ отгружен. Составлен акт о выполнении. Заказ закрыт»: заказ доставлен клиенту. Менеджер по продажам составляет акт о выполнении заказа, клиент подписывает акт. Менеджер по продажам закрывает заказ. Конечное состояние.

Состояние 9 «Заказ аннулирован»: заказ отменен по каким-либо причинам. Конечное состояние.

4.4.1.3. Описание состояний объекта класса «Employee» (сотрудник)

Состояние 1 «Неактивный сотрудник. Не предлагается к выбору в полях исполнителя. Запрещен вход по логину, привязанному к нему»: это новый сотрудник, или сотрудник находится в отпуске или на больничном. Уволившийся сотрудник не удаляется из базы данных, чтобы не нарушать существующие связи с другими таблицами. Неактивный сотрудник не может быть назначен для выполнения заказов либо другой работы. Новый сотрудник также является неактивным. Переход в состояние 2, если сотрудник активирован администратором, или объект остается в состоянии 1, если его не активировали. Начальное и конечное состояние.

Состояние 2 «Активный сотрудник. Может указываться в качестве исполнителя в полях выбора. Может осуществлять вход по логину, привязанному к нему»: сотрудник выполняет свою работу. Переход в состояние 1, если сотрудник уволился или ушел в отпуск, или объект остается в состоянии 2.

4.4.1.4. Описание состояний объекта класса «User» (пользователь)

Состояние 1 «Активный пользователь. Разрешен вход по логину» – это новый пользователь, или не деактивированный по каким-либо причинам пользователь. Переход в состояние 2, если сотрудник, на которого ссылается пользователь, уволился, или от сотрудника в общую сеть распространяются вирусы, или по иным причинам. Иначе объект остается в состоянии 1. Пользователь, привязанный к уволившемуся сотруднику, не удаляется из базы данных, чтобы не нарушать существующие связи с другими таблицами. Начальное и конечное состояние.

Состояние 2 «Неактивный пользователь. Запрещен вход по логину» – пользователь, привязанный к уволившемуся сотруднику, или деактивированный по определенным причинам пользователь. Пользователь, привязанный к временно отсутствующему (в отпуске или на больничном) сотруднику, не деактивируется. Переход в состояние 1, если администратор активирует пользователя повторно, или объект остается в состоянии 2.

4.4.1.5. Описание состояний объекта класса «AdMessage» (рекламное сообщение)

Состояние 1 «Черновик. Его можно удалить»: новое рекламное сообщение. Находится в процессе создания. Может быть удалено. Переход всегда в состояние 2. Начальное состояние.

Состояние 2 «Активное сообщение. Закончено редактирование и касается товаров, не снятых с производства. Может использоваться для рассылки»: сообщение для рассылки клиентам, содержит список активных товаров и актуальные цены. Редактирование сообщения невозможно. Переход в состояние 3, если товар снят с производства, или сообщение требуется отредактировать (например, обновить цены). Объект остается в состоянии 2, если сообщение не деактивировано.

Состояние 3 «Неактивное сообщение. Неактивированное сообщение или описывает товары, снятые с производства. Сообщение нельзя рассылать»: сообщение неактивно, возможно, временно. Сообщение в таком состоянии либо описывает товары, снятые с производства, либо сообщению требуется редактирование в связи с изменением цен на товары или с ошибкой в самом сообщении. Переход в состояние 2, если товары вернули в производство, или сообщение отредактировано и запущено. Остается в состоянии 3, если сообщение не активировали повторно. Конечное состояние.

4.4.1.6. Описание состояний объекта класса «Item» (товар)

Состояние 1 «Неактивное изделие. Не предлагается к выбору в заказах и заявках»: создан новый товар (изделие), либо товар временно снят с производства. Переход в состояние 2, если новый товар приняли в работу, или ранее снятый товар вернули в производство. Объект остается в состоянии 1, если он не запущен в производство. Начальное и конечное состояние.

Состояние 2 «Активное изделие. Может выбираться в заявках и заказах»: товар может быть добавлен в заказ клиентом. Переход в состояние 1, если цена на товар просрочена – т.е. цена на текущую дату не совпадает с ценой из журнала «Товар – Цена – Дата». Переход в состояние 1, если товар снимается с производства и деактивируется. Остается в состоянии 2, если пройдена проверка на актуальность цены и изделие не деактивировали.

4.5. Диаграмма развертывания (размещения)

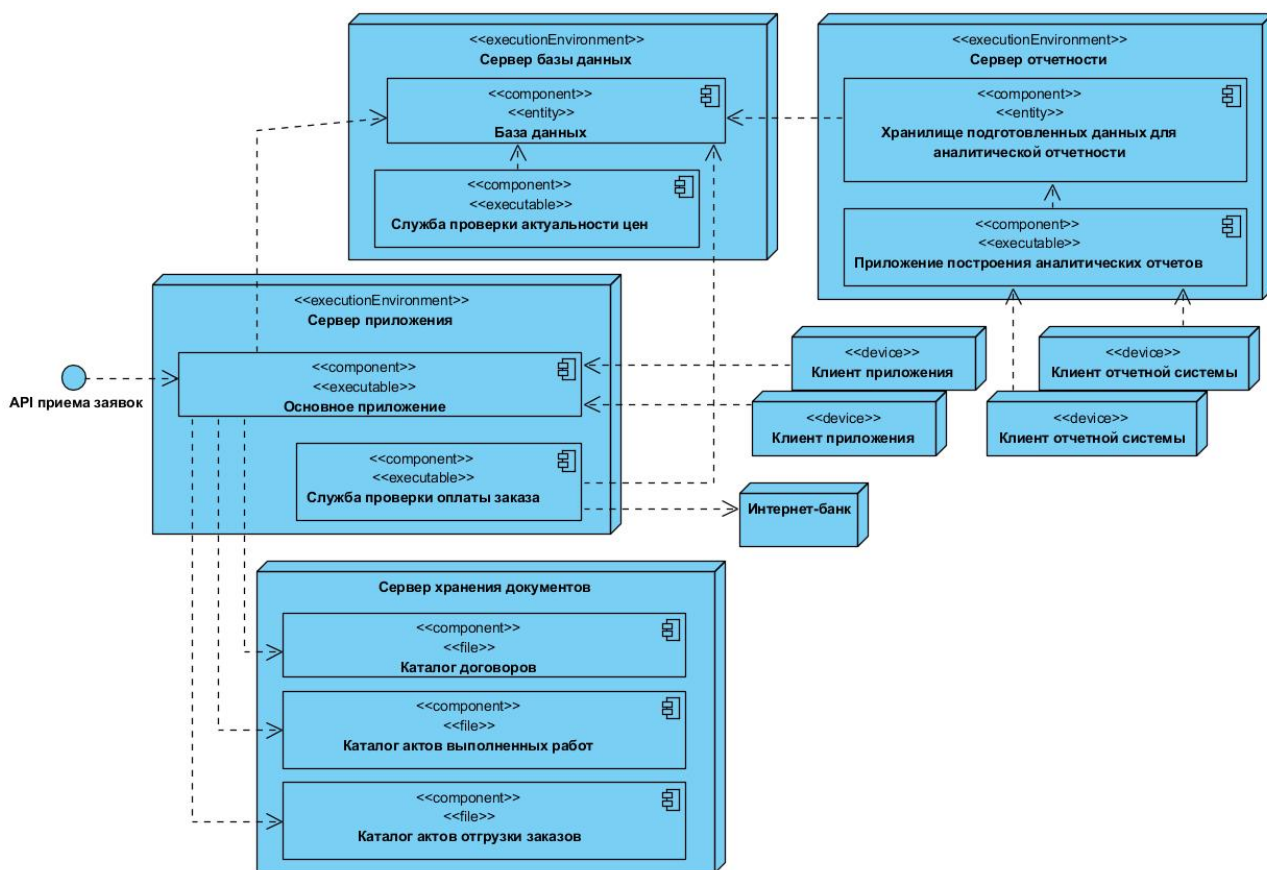


Рисунок 18 - Диаграмма размещения «Отдел сбыта предприятия»

4.5.1. Описание диаграммы размещения

Разработанная ИС физически располагается на 4 серверах.

4.5.1.1. Сервер приложения

На сервере приложения располагается основное приложение, через которое идет взаимодействие с базой данных. В приложении осуществляется работа с заказами, работа с каталогом изделий, работа с пользователями и ролями, на которые пользователь может быть назначен, работа с рекламными сообщениями и т.д.

Клиент приложения ставится на компьютеры менеджеров по продажам, системных администраторов, экономистов, специалистов технологического отдела, специалистов рекламного отдела, менеджеров по производству, начальников отделов, т.к. данные должности ведут работу через основное приложение, или имеют право смотреть на статус заказов, список пользователей и ролей и т.д. Компьютеров с поставленным клиентом основного приложения показано 2, но на самом деле их гораздо больше.

Также приложение имеет интерфейс «API приема заявок». С помощью этого API клиент может создать свое приложение либо сайт со своим интерфейсом, через который можно оставлять заявки на производство товаров.

На этом же сервере находится служба проверки оплаты заказа. Эта служба автоматически проверяет поступление оплаты от заказчика. Когда оплата за заказ поступает, служба оповещает об этом отдел бухгалтерии. Если в течение определенного времени (например, на оплату заказа дается неделя) оплата не поступила - служба автоматически аннулирует заказ (по timeout) и оповещает об этом менеджера по продажам, который вел заказ. Служба взаимодействует с БД, берет оттуда данные по заказу и может обновлять статус заказа. Служба также взаимодействует с приложением интернет-банка, располагающегося на компьютере бухгалтера, для проверки оплаты заказа. Компьютер с поставленным интернет-банком показан 1, но на самом деле их больше.

4.5.1.2. Сервер базы данных

На сервере базы данных содержится БД с информацией о заказах (журнал заказов), клиентах, сотрудниках, товарах, производимых на предприятии. В БД хранятся актуальные и неактуальные на данный момент данные. Большинство таблиц содержит дубликаты – history-таблицы с дополнительным полем типа DATE, в которые добавляется строка о любом изменении в основной таблице. Например, при обновлении телефона клиента в основной таблице телефон просто обновляется, а в history-таблицу добавляется новая строка с текущей датой (иногда и временем) и новым значением номера телефона. Такая организация позволяет хранить всю информацию за всю историю.

Также на сервере БД располагается служба проверки актуальности цен, используемая для контроля над ценами на товары. Эта служба по расписанию (например, раз в день) делает запрос в БД – достает данные из таблицы-журнала «Товар – цена – дата», которая была наполнена специалистами экономического отдела, и сверяет, соответствует ли действующая цена на товар в текущую дату цене из журнала актуальных цен. Если соответствия нет, служба временно деактивирует товар до редактирования цены на товар специалистами.

4.5.1.3. Сервер отчетности

Для отчетности выделен отдельный сервер.

Хранилище подготовленных данных для аналитической отчетности хранит в себе данные, скопированные из базы данных, в необходимом виде для создания отчетов по продажам и т.д.

Взаимодействие с хранилищем осуществляется через приложение построения аналитических отчетов. Приложение запрашивает и выдает подготовленные для составления отчетов данные, формирует из полученных данных отчет по продажам (либо другой отчет) и сохраняет отчет в файл.

Клиент приложения отчетной системы ставится на компьютеры менеджеров по

продажам, бухгалтеров, экономистов, юристов, начальников отделов, т.к. все эти должности имеют право запросить нужный отчет. Компьютеров с поставленным клиентом приложения отчетной системы показано 2, но на самом деле их гораздо больше.

4.5.1.4. Сервер хранения документов

Все документы в отсканированном виде удобнее хранить на отдельном сервере.

В каталоге договоров хранятся сканы договоров между заказчиком и компанией на изготовление указанного списка изделий по определенной цене до установленной даты. Сканы договоров добавляются из основного приложения менеджеров по продажам.

В каталоге актов выполненных работ хранятся сканы этих актов, добавленные менеджеров по продажам из основного приложения.

Каталог актов отгрузки заказов хранит информацию о факте отгрузки заказов с производства, добавляется менеджером по производству через основное приложение.

Любой акт может быть запрошен через основное приложение сотрудниками компании.