Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

«Уфимский государственный нефтяной технический университет»

Кафедра цифровых технологий и моделирования

**Курсовая работа**

**По дисциплине:** Проектирование программного обеспечения

|  |  |
| --- | --- |
| Выполнила: студент гр. БПИз-20-01 | Дмитриева Е.К. |
|  | (дата, подпись) |
| Проверила: доцент | Давлиева А.С. |
|  | (дата, подпись) |

**Уфа - 2023**

**Оглавление**

[Задание на курсовую работу 3](#_Toc150871631)

[Описание предметной области 5](#_Toc150871632)

[Функциональное моделирование 6](#_Toc150871633)

[Решение для автоматизации бизнес-процессов с использованием программного продукта 8](#_Toc150871634)

[Система требований к создаваемому программному продукту и атрибуты качества 9](#_Toc150871635)

[Диаграмма Вариантов использования 13](#_Toc150871636)

[Диаграмма классов 14](#_Toc150871637)

[Диаграмма последовательности 15](#_Toc150871638)

[Диаграмма состояний 16](#_Toc150871639)

[Диаграмма активности 17](#_Toc150871640)

[Диаграмма взаимодействия 18](#_Toc150871641)

[Диаграмма компонентов 19](#_Toc150871642)

[Диаграмма развертывания 20](#_Toc150871643)

[Система тестов программного продукта 21](#_Toc150871644)

[Шаблон отчета о проведении тестирования 24](#_Toc150871645)

[Список использованной литературы 26](#_Toc150871646)

# **Задание на курсовую работу**

**по дисциплине «Проектирование программного обеспечения»**

**2022/2023 учебный год**

1. Изучить предоставленное описание предметной области.
2. Построить модель бизнес-процессов указанной предметной области с использованием технологии функционального моделирования IDEF0
3. Предложить решение для автоматизации бизнес-процессов с использованием программного продукта.
4. Разработать систему требований к создаваемому программному продукту и атрибуты качества:
   1. бизнес-требования;
   2. функциональные требования;
   3. требования к интерфейсу;
   4. требования к надежности работы;
   5. системные требования и требования к совместимости с внешними программными средствами.
5. Оформить требования в виде соответствующих UML-диаграмм.
6. Разработать иерархию классов программного продукта и представить ее в виде диаграммы классов
7. Описать работу и размещение программного продукта с использованием следующих UML-диаграмм:
   1. диаграмма последовательности (Sequence Diagram);
   2. диаграмма состояний (State Diagram);
   3. диаграмма активности (Activity Diagram);
   4. диаграмма взаимодействия (Cooperation Diagram);
   5. диаграмма компонентов (Component Diagram);
   6. диаграмма развертывания (Deployment Вiagram).
8. Разработать систему тестов программного продукта для проверки реализации всех требований и соответствия атрибутам качества.
9. Разработать шаблон отчета о проведении тестирования.
10. Оформить итоговый проект программного продукта в виде пояснительной записки к курсовой работе (формат Word, OpenOffice Write, LibreOffice Write). Помимо пунктов описания проекта, пояснительная записка должна также содержать:
    1. титульный лист;
    2. оглавление;
    3. текст задания;
    4. список использованной литературы (не менее 5 источников), оформленный в соответствии с ГОСТ Р7.0.100-2018 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления».

# **Описание предметной области**

Основной задачей магазина является поставка клиентам музыкальных инструментов в соответствии с их требованиями. Поскольку большая часть инструментов поставляется из-за рубежа, магазин действует в соответствии с несколькими документами: закон о защите прав потребителя, закон о торговой деятельности, закон о таможенном регулировании.

Продавцы-консультанты собирают заявки и уточняют у клиентов их требования, желаемый срок поставки и стоимость товара.

Сотрудники отдела закупок осуществляют контакты с фирмами производителями (заказывают товар), с транспортными компаниями и таможенной службой РФ. Таким образом, отдел закупок полностью управляет всем процессом перемещения товара от производителя до склада компании.

Сотрудники склада принимают товар, регистрируют его наличие и договариваются с клиентами о его доставке до конечной точки.

Всеми финансовыми вопросами занимается финансовый департамент.

# **Функциональное моделирование**

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой «черный ящик» со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня.

Функциональная модель представлена на рисунках 1-2.

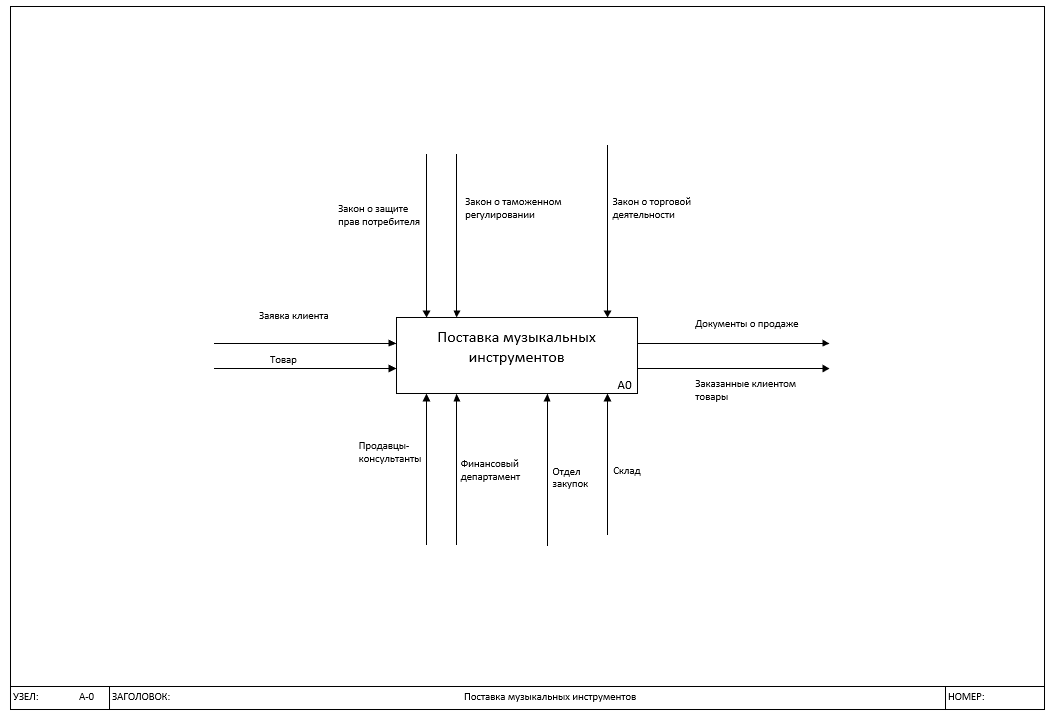


Рисунок 1 - Уровень 0. Общий контекст магазина музыкальных инструментов

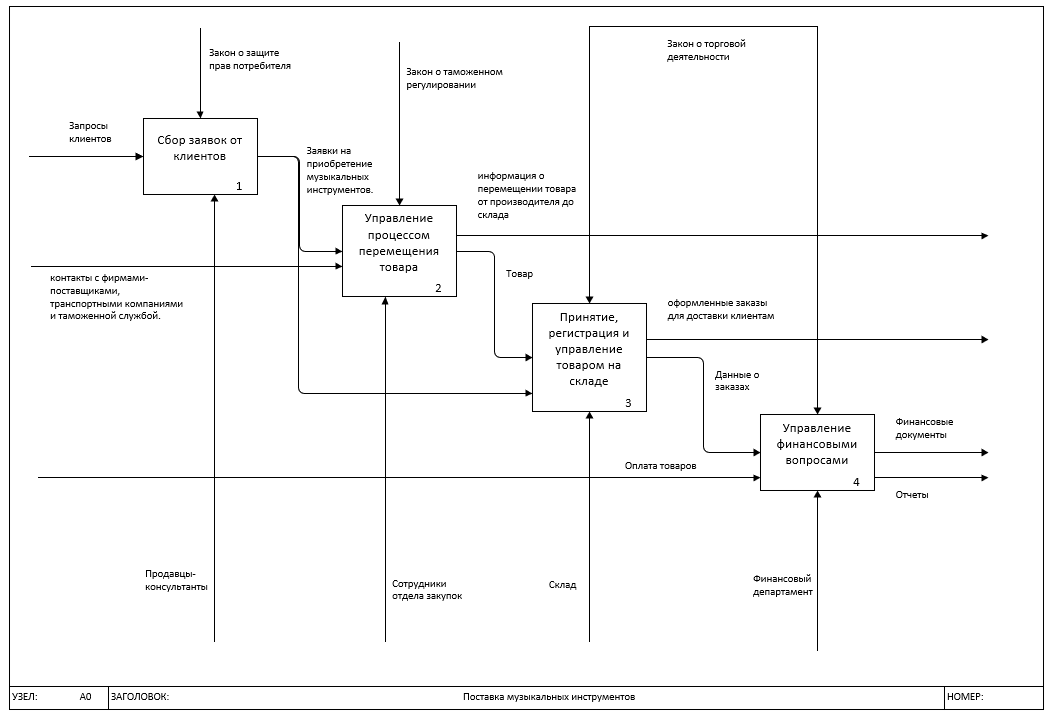


Рисунок 2 - Уровень 1. Функциональные блоки

# **Решение для автоматизации бизнес-процессов с использованием программного продукта**

Для автоматизации бизнес-процессов магазина, нужно разработать программный продукт, который будет интегрировать различные функции и отделы компании.

Приложение должно решать следующие задачи и функции :

1. включать сбор заявок от клиентов;
2. управление заказами;
3. отслеживание поставок;
4. учет инвентаря;
5. финансовое управление и отчетность.

Внедрение программного продукта для автоматизации бизнес-процессов позволит улучшить эффективность работы, уменьшить вероятность ошибок и повысить удовлетворенность клиентов, следуя законодательству и соблюдая стандарты качества.

# **Система требований к создаваемому программному продукту и атрибуты качества**

Бизнес-требования.

Для магазина, который занимается продажей музыкальных инструментов, существуют различные бизнес-требования, которые помогут обеспечить эффективное функционирование и соответствие законодательству. Вот некоторые из них:

1. Управление заказами и клиентскими запросами:
   1. Система управления заказами и запросами должна быть реализована для эффективного сбора и обработки заявок от клиентов.
   2. Документирование и сохранение запросов клиентов для обеспечения соблюдения закона о защите прав потребителя.
   3. Отслеживание статусов заказов для обеспечения своевременной доставки.
2. Управление поставками и инвентаризацией:
   1. Система управления складом и поставками должна обеспечивать связь с производителями, транспортными компаниями и таможенной службой.
   2. Автоматизированная система для заказа и отслеживания поставок товаров.
   3. Мониторинг инвентаря на складе, чтобы удовлетворить спрос клиентов и избегать излишков или дефицита товаров.
3. Соблюдение законодательства:
   1. Следование закону о защите прав потребителя в отношении возврата и обмена товаров, гарантий, и др.
   2. Соблюдение закона о торговой деятельности, включая требования к регистрации бизнеса и уплате налогов.
4. Учет и финансы:
   1. Автоматизированная бухгалтерская система для учета всех финансовых операций, включая оплату поставщикам, обработку заказов и расчеты с клиентами.
   2. Подготовка финансовых отчетов для внутреннего управления и соблюдения закона о финансовой отчетности.

Функциональные требования.

Управление клиентскими данными и заказами:

1. Регистрация клиентов и хранение их контактной информации.
2. Возможность создания и редактирования заказов, включая выбор инструментов, количество, цены и сроки доставки.
3. Отслеживание истории заказов клиентов.

Управление складом:

1. Учет и отслеживание инвентаря на складе, включая входящие поставки и продажи.
2. Система оповещения при нехватке товара на складе.

Управление закупками:

1. Возможность связи с поставщиками и фирмами-производителями.
2. Оформление заказов у поставщиков.
3. Отслеживание статусов поставок.

Управление логистикой и доставкой:

1. Отслеживание статусов доставки товаров клиентам.
2. Интеграция с транспортными компаниями для эффективной логистики.

Финансовое управление:

1. Возможность учета финансовых операций, включая оплату заказов клиентами и оплату поставщикам.
2. Отчеты и анализ финансовых данных.

Требования к интерфейсу.

Исходя из описанных процессов в магазине, интерфейс системы должен быть удобным, интуитивно понятным и эффективным для использования всеми сотрудниками.

Пользовательский интерфейс для продавцов-консультантов:

Простота использования: Интерфейс должен быть легко понимаемым, с минимальным количеством сложных элементов управления.

Быстрый доступ к информации: Консультанты должны иметь быстрый доступ к каталогу товаров, а также к информации о наличии на складе и статусе заказов.

Функции для обработки заявок: Модуль для сбора заявок от клиентов с возможностью внесения необходимых деталей, таких как предпочтения и контактные данные.

Интерфейс для отдела закупок:

Система управления поставками: Интерфейс должен позволять сотрудникам отслеживать статус заказов, связываться с поставщиками и контролировать перемещение товаров.

Управление складскими запасами: Доступ к информации о текущем состоянии склада, поступлении товаров и их распределении по категориям.

Интерфейс для сотрудников склада:

Прием товара: Функции для регистрации поступления товара, проверки его качества и актуализации информации о наличии товаров на складе.

Управление доставкой: Инструменты для планирования и отслеживания доставки товаров клиентам, включая контакт с транспортными компаниями.

Интерфейс для финансового департамента:

Управление финансами: Возможность отслеживания финансовых операций, учета расходов и доходов, а также генерации финансовых отчетов.

Общие требования:

Безопасность: Система должна обеспечивать безопасность данных и иметь возможность управления правами доступа для различных уровней пользователей.

Мобильная совместимость: Возможность доступа к системе через мобильные устройства для работы вне офиса.

Отчетность и аналитика: Возможность генерации отчетов о продажах, остатках на складе, популярности товаров и других ключевых метриках.

Разработка такой системы должна быть ориентирована на автоматизацию бизнес-процессов, улучшение эффективности работы сотрудников и обеспечение качественного обслуживания клиентов.

# **Диаграмма Вариантов использования**

В этой диаграмме описывается, что делает система, но не то, как она это делает. Вариант использования — это набор событий, которые происходят, когда «оператор» использует систему для завершения процесса. Оператор определяется как кто-либо или что-либо, взаимодействующее с системой (человек, организация или приложение) из-за пределов системы.

Таким образом, диаграмма вариантов использования визуально описывает этот набор последовательностей и представляет функциональные требования системы.

Диаграмма вариантов использования представлена на рисунке 3.

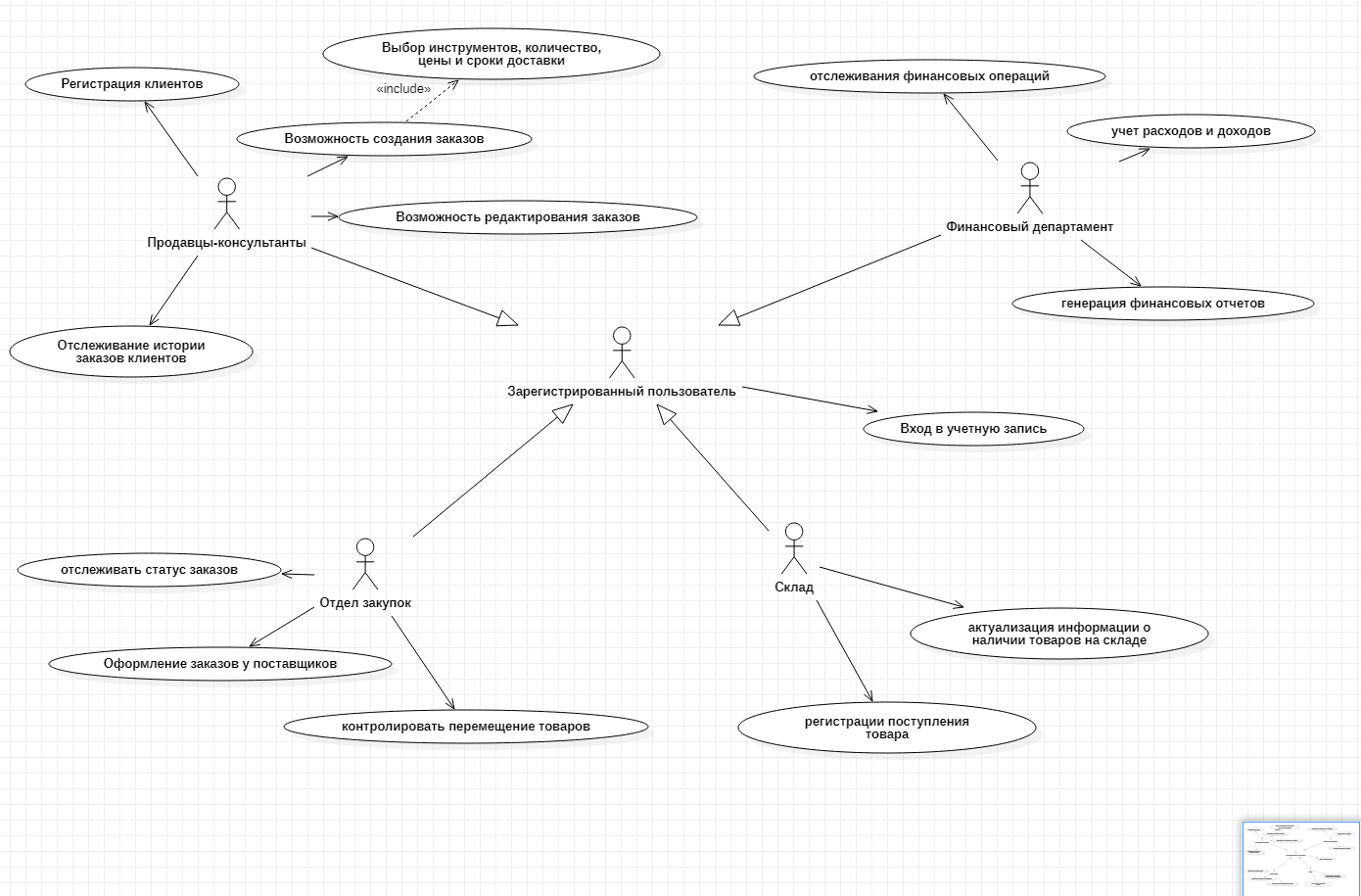


Рисунок 3 -Диаграмма вариантов использования

# **Диаграмма классов**

Эта диаграмма, наиболее распространенная при разработке ПО, используется для изображения логической и физической структуры системы и показывает ее классы. Она похожа на блок-схему, потому что классы представлены в виде блоков. Эта диаграмма предлагает визуальное представление о различных классах и о том, как они взаимосвязаны.

Диаграмма классов представлена на рисунке 4.

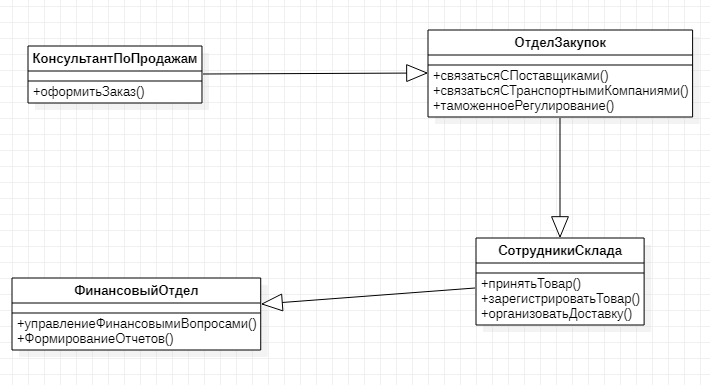


Рисунок 4 - Диаграмма классов

# **Диаграмма последовательности**

Эта визуально привлекательная диаграмма, популярная не только в сообществе разработчиков, хорошо показывает все типы бизнес-процессов. Она просто раскрывает структуру системы, показывая последовательность сообщений и взаимодействий между операторами и объектами в хронологическом порядке. Диаграммы последовательности отображают простую итерацию и ветвление.

Диаграмма последовательности представлена на рисунке 5.

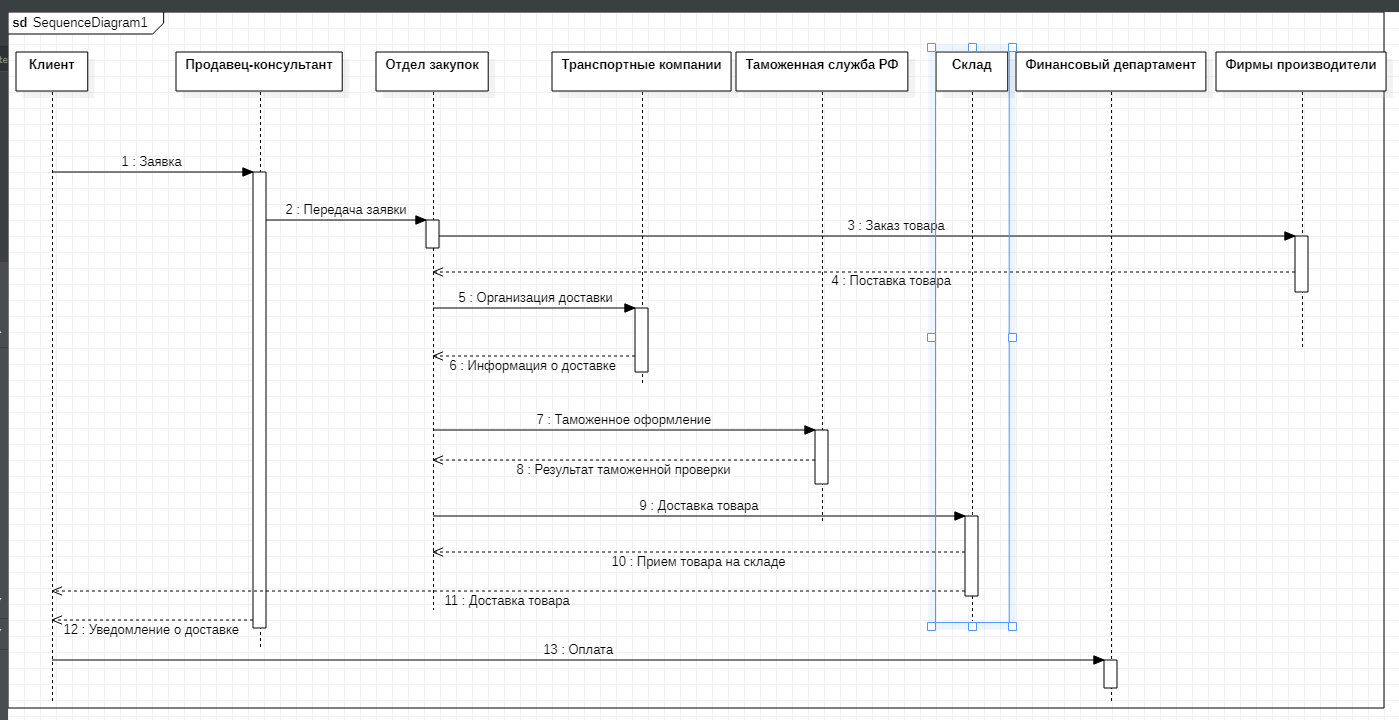


Рисунок 5 - Диаграмма последовательности

# **Диаграмма состояний**

В отличие от других диаграмм, диаграмма состояний описывает процесс изменения состояний только одного класса, а точнее — одного экземпляра определенного класса, т. е. моделирует все возможные изменения в состоянии конкретного объекта на протяжении всего его жизненного цикла.

Диаграмма состояний представлена на рисунке 6.

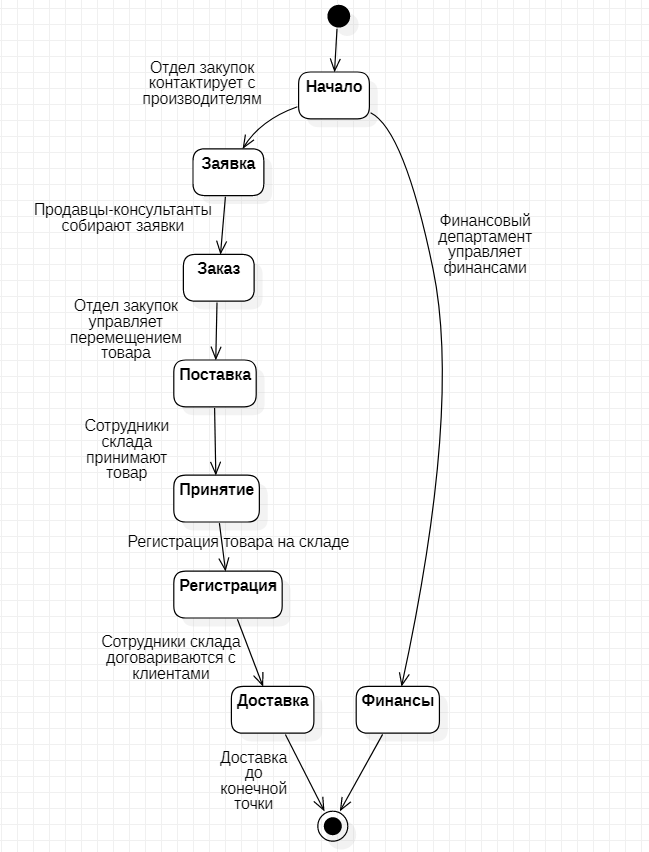
ы

Рисунок 6 - Диаграмма состояний

# **Диаграмма активности**

Диаграмма активностей (видов деятельности) — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний.

Диаграмма активности показа на рисунке 7.

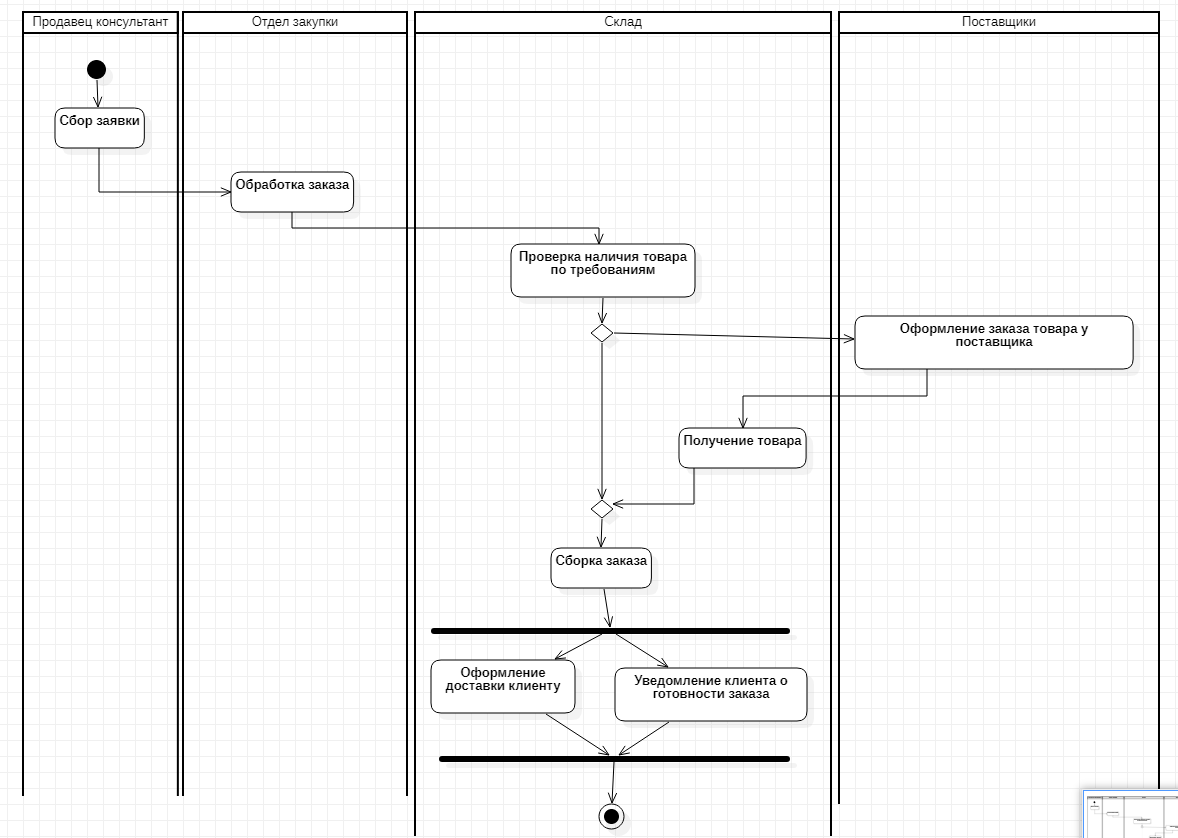


Рисунок 7 - Диаграмма активности

# **Диаграмма взаимодействия**

К диаграммам взаимодействия относятся диаграммы кооперации и диаграммы последовательности.

Диаграммы коммуникации (раньше они назывались диаграммами кооперации) показывают связи между объектами. По этим диаграммам можно судить, как объекты взаимодействуют друг с другом посредством сообщений в пределах архитектуры системы.

Диаграмма коммуникации представлена на рисунке 8.

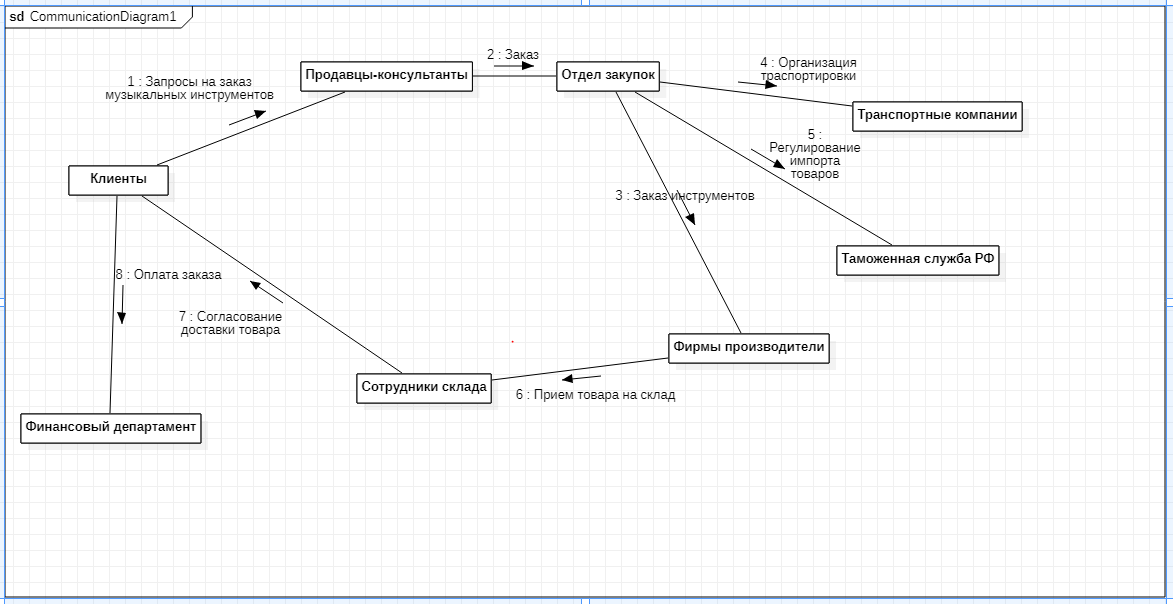


Рисунок 8 - Диаграмма коммуникаций

# **Диаграмма компонентов**

Диаграммы компонентов используются для визуализации организации компонентов системы и зависимостей между ними. Они позволяют получить высокоуровневое представление о компонентах системы.

Компонентами могут быть программные компоненты, такие как база данных или пользовательский интерфейс; или аппаратные компоненты, такие как схема, микросхема или устройство; или бизнес-подразделение, такое как поставщик, платежная ведомость или доставка.

Диаграмма активности показа на рисунке 9.

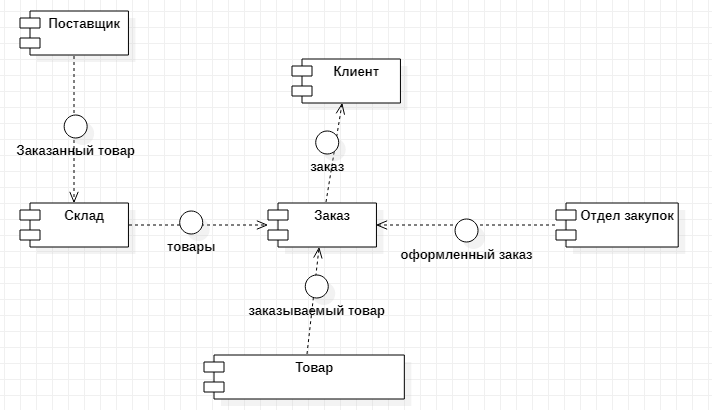


Рисунок 9 - Диаграмма компонентов

# **Диаграмма развертывания**

Диаграмма развертывания (Deployment Diagram) — это вид диаграммы UML, который моделирует физическое размещение компонентов системы и их взаимосвязи на вычислительных узлах (например, компьютерах, серверах или устройствах). Она предоставляет визуальное представление архитектуры системы, фокусируясь на размещении элементов и их взаимодействии в реальной физической среде.

Диаграмма развертывания показа на рисунке 10.

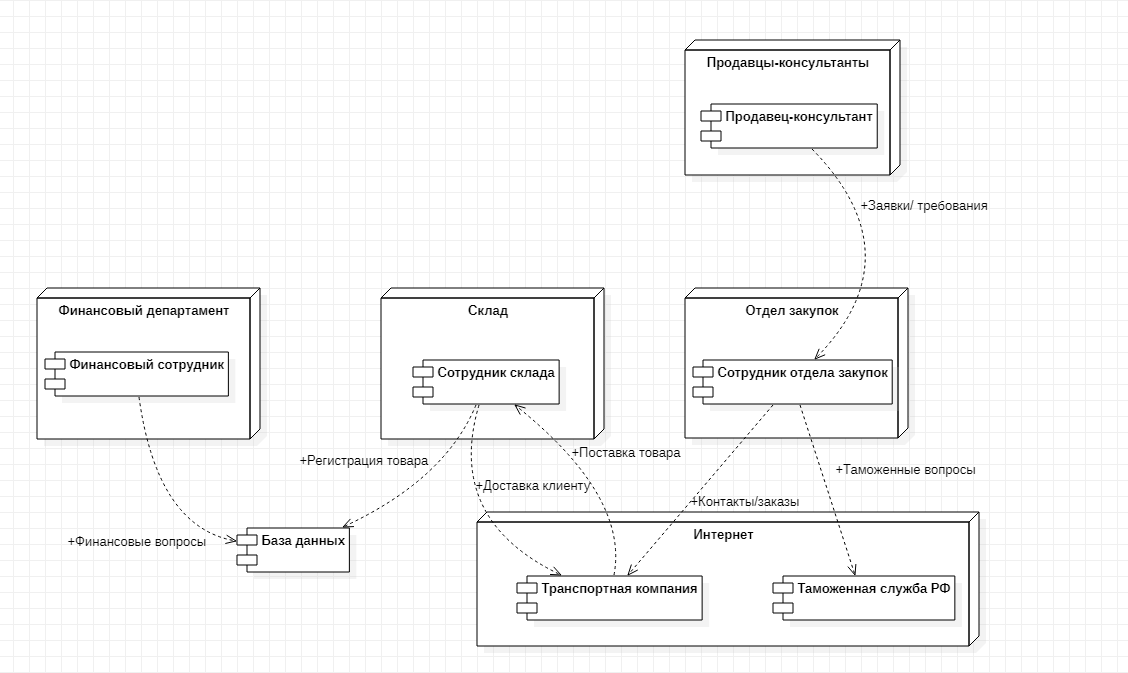


Рисунок 10 - Диаграмма развертывания

# **Система тестов программного продукта**

Для разработки системы тестов программного продукта, необходимо определить тестовые случаи, которые позволят проверить реализацию всех требований и соответствие атрибутам качества в данной предметной области. Вот примеры тестовых случаев для различных компонентов системы:

1. Продавцы-консультанты:
   1. Тестирование сбора заявок и требований от клиентов.
   2. Проверка корректности обработки информации о желаемом сроке поставки и стоимости товара.
   3. Тестирование взаимодействия с клиентами для уточнения их требований.
2. Отдел закупок:
   1. Тестирование контактов с фирмами производителями и корректного размещения заказов.
   2. Проверка взаимодействия с транспортными компаниями и таможенной службой РФ.
   3. Тестирование управления процессом перемещения товара от производителя до склада компании.
3. Сотрудники склада:
   1. Тестирование процесса приемки товара на склад и его регистрации в системе.
   2. Проверка взаимодействия с клиентами для договоренности о доставке товара до конечной точки.
4. Финансовый департамент:
   1. Тестирование системы управления финансовыми вопросами, включая расходы, доходы, и обработку финансовых данных.

Обобщенные шаблоны тестирования:

1. Продавцы-консультанты
   1. Тестирование сбора заявок:
      1. Заполнение формы заявки с разными комбинациями данных.
      2. Проверка корректного сохранения информации о заявке в базе данных.
   2. Тестирование уточнения требований:
      1. Взаимодействие с системой для уточнения требований клиента.
      2. Проверка, что полученная информация правильно связана с соответствующей заявкой.
   3. Тестирование взаимодействия с клиентами:
      1. Сценарии взаимодействия с клиентами через различные каналы (например, телефон, электронная почта).
      2. Проверка корректного обновления статуса заявки после общения с клиентом.
2. Отдел закупок:
   1. Тестирование контактов с фирмами производителями:
      1. Создание заказа у поставщика и проверка его успешной регистрации.
      2. Сценарии обработки изменений в заказе.
   2. Тестирование взаимодействия с транспортными компаниями и таможенной службой РФ:
      1. Проверка правильности передачи информации о грузе транспортным компаниям.
      2. Тестирование обработки данных в соответствии с требованиями таможенного регулирования.
3. Сотрудники склада:
   1. Тестирование приемки товара на склад:
      1. Симуляция приемки товара с различными характеристиками.
      2. Проверка автоматической регистрации товара в базе данных.
   2. Тестирование взаимодействия с клиентами:
      1. Взаимодействие с системой для уточнения деталей доставки товара.
      2. Тестирование процесса обновления статуса доставки.
4. Финансовый департамент:
   1. Тестирование системы управления финансовыми вопросами:
      1. Ввод финансовых транзакций и проверка корректности их обработки.
      2. Тестирование генерации отчетов и анализа финансовых данных.
5. Взаимодействие между компонентами:
   1. Тестирование передачи данных между компонентами:
      1. Проверка правильности передачи заказов от продавцов-консультантов к отделу закупок.
      2. Тестирование обновления данных о товарах и статусах заказов.
6. Атрибуты качества:
   1. Тестирование производительности:
      1. Сценарии с множеством одновременных пользователей для оценки производительности системы.
      2. Проверка времени ответа системы в различных условиях нагрузки.
   2. Тестирование безопасности:
      1. Проверка системы на наличие уязвимостей безопасности.
      2. Тестирование правильности работы механизмов аутентификации и авторизации.
   3. Тестирование отказоустойчивости:
      1. Сценарии тестирования восстановления после сбоев и проверка сохранности данных.

# **Шаблон отчета о проведении тестирования**

Отчет о проведении тестирования — это документ, который предоставляет обзор выполненных тестов, результатов, обнаруженных ошибок и другой важной информации о качестве программного продукта. Ниже представлен общий шаблон отчета о проведении тестирования:

Таблица 1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта |  |
| Номер версии |  |
| Имя тестера |  |
| Даты тестирования |  |

Таблица 2 – Описание информационных полей для тестирования

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | Уникальный ID для каждого testcase. Следуйте определенной логике именования и нумерации. например ‘TC\_UI\_1′ указание на пользовательский интерфейс testcase #1′. |
| Приоритет тестирования *(Малый/Средний/высокий)* | Насколько важен каждый тест. Приоритет при испытании бизнес-правил или функционала может быть средним или высоким, в то время как незначительные формы пользовательского интерфейса могут быть с низким приоритетом. |
| Название тестирования/ Имя | Название тестирования. Например, проверка формы авторизации с правильным логином и паролем. |
| Резюме испытания | Описание, чего нужно достигнуть при тестировании. |
| Шаги тестирования | Перечислите детально все шаги тестирования. Напишите в каком порядке должны быть выполнены эти шаги. Убедитесь, что вы обеспечили настолько максимальную детализацию насколько можете. Нумерованный список – будет хорошей идей |
| Данные тестирования | Напишите тестовые данные, используемые для этого тестирования. Таким образом, актуальные данные, которые будут предложены и будут использоваться для проведения тестирования. Например, логин и пароль – для входа в систему. |
| Ожидаемый результат | Какой должен получится результат после выполнения теста? Опишите подробно ожидаемый результат, включая любые сообщения и ошибки, которые должны быть, выданы на экран. |
| Фактический результат | Какой фактический результат после выполнения теста? Опишите любое соответствующее поведение системы после выполнения тестирования. |
| Предпосылки | Любые предварительные действия, которые должны быть выполнены перед проведением тестирования. Перечислите предварительные условия, для успешного выполнения проекта |
| Постусловия | Какое состояние должно быть у системы после выполнения тестирования? |
| Статус *(Pass/Fail)* | Еслифактическийрезультатнесоответствуетожидаемымрезультатамотметка, что тест провалился (fail). В противном случае как прошло (pass) |

Продолжение таблицы 2

|  |  |
| --- | --- |
| Наименование | Описание |
| Комментарии | Используйте эту область для любых дополнительных записей или комментариев. Это область нужна для поддержки полей выше (например, есть какие-то особые условия, которые не могут быть описаны ни в одном из полей или есть вопросы, связанные с ожидаемыми или фактическими результатами) |

# **Список использованной литературы**

1. ГОСТ Р 7.0.100-2018. Библиографическая запись. Библиографическое описание. Общие требования и правила составления : национальный стандарт Российской Федерации : дата введения 2019-07-01 / Федеральное агентство по техническому регулированию. – Изд. официальное. – Москва : Стандартинформ, 2018. – 124 с.
2. Грекул, В. И. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — Москва : Издательство Юрайт, 2021. — 385 с. — (Профессиональное образование). — ISBN 978-5-534-12104-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/476534 (дата обращения: 10.11.2023).
3. Простое руководство по UML-диаграммам и моделированию баз данных// Майкрософт: [сайт]. – 2019. – URL: https://www.microsoft.com/ru-ru/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling (дата обращения: 11.11.2023).
4. Маслеников К.Ю., Ревунков Г.И., Сатова М.В. Описание предметной области как неотъемлемый элемент процесса проектирования автоматизированной информационной системы // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 9, №6 [сайт]. – 2017. – URL: <https://naukovedenie.ru/PDF/55TVN617.pdf> (дата обращения: 10.11.2023).
5. Александр Марголин, UML для бизнес-моделирования: зачем нужны диаграммы процессов // Evergreen : [сайт]. – 2021. – URL: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> (дата обращения: 11.11.2023).