**ВВЕДЕНИЕ**

Современные условия ведения внешнеэкономической деятельности требуют от предприятий высокой эффективности таможенного администрирования. Одним из ключевых факторов успешного взаимодействия с таможенными органами является грамотное управление документами: от их регистрации и проверки до формирования отчетности и электронного обмена с государственными системами. Однако многие компании до сих пор сталкиваются с проблемами, связанными с ручной обработкой данных, отсутствием интеграции между информационными системами и сложностями контроля соответствия законодательным требованиям. Это приводит к задержкам в таможенном оформлении, финансовым потерям из-за штрафов и снижению общей эффективности логистических процессов.

Актуальность темы дипломного проекта обусловлена необходимостью автоматизации процессов работы с таможенной документацией для повышения скорости обработки данных, минимизации ошибок и обеспечения юридической значимости документооборота. Внедрение специализированного программного решения позволит унифицировать процессы регистрации, согласования и хранения документов, а также обеспечит прозрачность всех этапов таможенного оформления.

Целью работы является разработка информационной системы для автоматизации ведения таможенных документов, включая их регистрацию и проверку. Для достижения этой цели необходимо решить следующие задачи:

1. Провести анализ существующих решений и выявить их недостатки.
2. Определить функциональные требования к системе.
3. Разработать архитектуру и модульную структуру ИС.
4. Реализовать ключевые модули.
5. Обеспечить безопасность данных и контроль доступа.
6. Оценить экономическую эффективность внедрения системы.

**1 АНАЛИЗ ОБЪЕКТА АВТОМАТИЗАЦИИ**

**1.1 Структура объекта автоматизации**

На рисунке 1.1 показана структура предприятия «Белтаможсервис». Руководящий состав состоит из Генерального директора, технического директора и его заместителя, начальников отделов, подразделений и служб и их заместителей. Заместитель генерального директора по цифровизации контролирует работу офиса цифровизации. В состав этого офиса входит несколько отделов. Отдел программного обеспечения занимается созданием и сопровождением программ для предприятия. Отдел тестирования занимается тестированием программного обеспечения. Аналитический отдел занимается проектированием продукта и анализом что нужно изменить в процессах или в системе управления, чтобы увеличить эффективность предприятия. Бюро автоматизации технологий занимается налаживаем коммуникации, разработкой и настройкой инструментов для совместной работы.

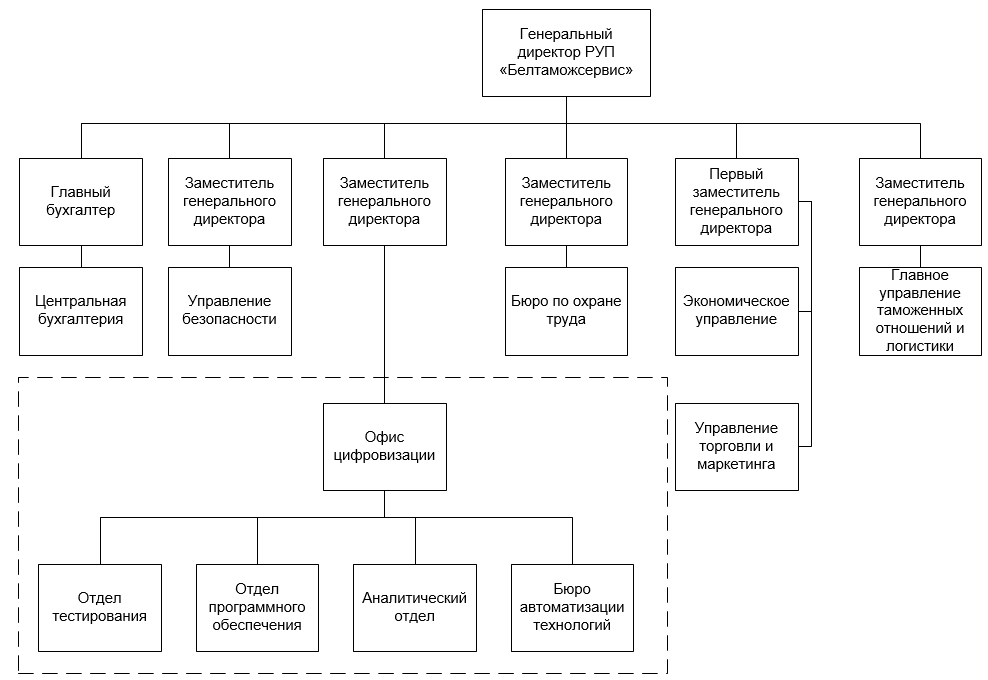


Рисунок 1.1 – Структура предприятия

**1.2 Анализ информационной системы**

Существующая ИС представляет собой комплекс взаимосвязанных подсистем, каждая из которых отвечает за определенный этап процесса таможенного оформления. Основные функциональные модули включают подсистему электронного декларирования, модуль расчета таможенных платежей, систему управления рисками и аналитический блок. Технологический стек существующей системы включает серверные компоненты на Java Spring Boot, клиентские приложения на платформе Vue.js и СУБД Oracle SQL.

Анализ выявил несколько критических недостатков текущей реализации. Основные проблемы включают устаревший пользовательский интерфейс, недостаточную производительность при обработке больших объемов данных и ограниченные возможности масштабирования.

На основании проведенного анализа определены ключевые направления модернизации. Приоритетными задачами являются переход на современные веб-технологии и оптимизация процессов обработки данных. Особое значение имеет разработка гибкой системы интеграционных адаптеров для упрощения подключения к новым внешним сервисам.

**1.3 Обзор аналогов**

На этапе проектирования системы были тщательно изучены существующие аналоги.

**1.3.1 Национальная автоматизированная система электронного декларирования**

Национальная автоматизированная информационная система электронного декларирования (НАСЭД) – система, осуществляющая информационную поддержку и автоматизацию таможенных операций, совершаемых должностными лицами таможенных органов и заинтересованными лицами (декларантами), с использованием письменных и электронных документов, а также обеспечивающая информационное взаимодействие таможенных органов Республики Беларусь с заинтересованными лицами и таможенными службами иных государств. НАСЭД является самостоятельной системой, входящей в состав единой автоматизированной информационной системы таможенных органов Республики Беларусь. Внедрение НАСЭД позволило сократить временные и финансовые затраты, минимизировать влияния субъективного фактора, ошибок или злоупотребления при проведении таможенных операции, а также способствовало предотвращению коррупции.

Целями и задачами этой информационной системы являются:

– сокращение времени таможенных операций на выпуск товаров за счет электронного обмена информацией между участниками внешнеэкономической деятельности и таможенными органами Республики Беларусь;

– внедрение на практике заявительного принципа, заключающегося в выпуске товаров на основании сведений, содержащихся в электронной таможенной декларации и электронном документе таможенного транзита; – прозрачность процесса совершения таможенных операций в отношении товаров;

– сокращение временных и финансовых затрат участников внешнеэкономической деятельности, связанных с таможенными операциями;

– предотвращение коррупции (исключение личного контакта должностного лица таможенного органа и участников внешнеэкономической деятельности);

– минимизация влияния субъективного фактора, ошибок или злоупотреблений при проведении таможенных операций.

Схема взаимодействия системы показана на рисунке 1.2.



Рисунок 1.2 – Информационная система НАСЭП

**1.3.2 *VED1***

*VED1* — это информационная система, разработанная для автоматизации процессов, связанных с внешнеэкономической деятельностью в Республике Беларусь. Она служит для учета и обработки данных о внешнеэкономических операциях, а также для взаимодействия с таможенными органами. Основные функции *VED1*:

– *VED1* позволяет пользователям регистрировать и отслеживать все внешнеэкономические сделки, включая экспорт и импорт товаров;

– cистема автоматически генерирует необходимые документы для таможенного оформления, что сокращает время на подготовку и минимизирует вероятность ошибок;

– предоставляет инструменты для анализа данных о внешнеэкономической деятельности, что позволяет пользователям получать актуальную информацию о рынке и тенденциях;

– *VED1* может интегрироваться с другими информационными системами, что позволяет улучшить обмен данными между различными государственными и частными структурами;

– система включает в себя функции поддержки пользователей, позволяя быстро получать ответы на вопросы и решать возникающие проблемы.

**1.3.3 *SAP Global Trade Services***

*SAP Global Trade Services* (*GTS*) — это мощное решение для управления внешнеэкономической деятельностью и таможенными процессами, разработанное компанией *SAP* для стран Европейского союза. Оно предназначено для автоматизации и оптимизации процессов импорта и экспорта, а также для обеспечения соблюдения международных торговых норм и требований. Интерфейс приложения показан на рисунке 1.3.

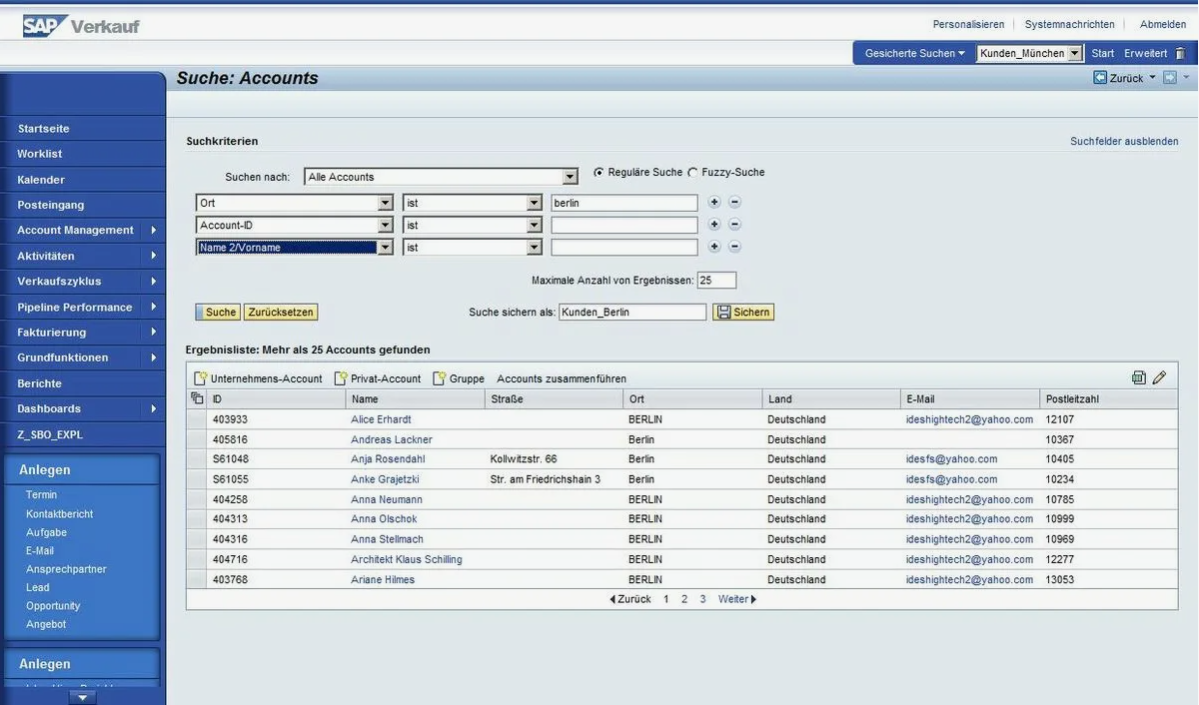


Рисунок 1.3 – Информационная система *SAP* *GTS*

Преимущества использования *SAP* *GTS*:

– система значительно сокращает время на оформление документов и минимизирует вероятность ошибок, связанных с ручным вводом данных;

– помогает компаниям соблюдать все необходимые законодательные и нормативные требования, что снижает риски штрафов и санкций;

– интеграция с другими бизнес-процессами и системами обеспечивает полную видимость операций и помогает лучше управлять внешнеэкономической деятельностью;

– адаптирована под специфические потребности компании и легко масштабируется по мере роста бизнеса.

Недостатки этой системы:

– внедрение этой информационной системы является дорогостоящим процессом, включая затраты на лицензии, оборудование, обучение и консультационные услуги;

– процесс внедрения может занять много времени, что может вызвать задержки в использовании системы и потребовать от компании дополнительных ресурсов.

* + 1. **Автоматизированная Информационная Система Таможни**

АИСТ (Автоматизированная Информационная Система Таможни) – это централизованная государственная система, обеспечивающая электронное декларирование, контроль и управление таможенными процессами в России. Она является основой цифровизации таможенных операций и интегрирована с другими информационными системами Федеральной таможенной службы (ФТС России) и Евразийского экономического союза (ЕАЭС).

Система АИСТ выполняет следующие задачи: электронное декларирование товаров, автоматизированный контроль и выпуск товаров, управление таможенными платежами, обмен данными с участниками ВЭД (брокерами, перевозчиками, экспортерами и импортерами).

Преимущества этой системы:

– совместимость с системами ЕАЭС;

– возможность предварительного расчета пошлин;

– прозрачность всех операций (платежи, декларирование, выпуск товаров);

– минимизация личного контакта между участниками ВЭД и таможенникам;

– система управления рисками выявляет подозрительные поставки с помощью алгоритмов.

Недостатки этой системы:

– ложные срабатывания системы управления рисками, ведущие к задержкам грузов;

– запаздывание обновлений системы под новые нормы;

– сбои в работе системы при высокой нагрузке.

**1.3.5 Выводы**

Все рассмотренные системы демонстрируют единую направленность на цифровизацию таможенного администрирования, однако имеют существенные различия в функциональных возможностях и архитектурных решениях.

Национальная автоматизированная система электронного декларирования (НАСЭД) успешно реализует заявительный принцип таможенного оформления, обеспечивая прозрачность всех операций. Однако архитектура системы ориентирована преимущественно на национальные потребности, что может ограничивать ее адаптацию к международным стандартам.

Информационная система VED1 демонстрирует узкоспециализированный подход к автоматизации внешнеэкономической деятельности. Ее ключевое преимущество заключается в глубокой проработке функций аналитики и генерации документов. Однако ограниченная интеграционная способность и ориентация исключительно на белорусский рынок снижают потенциал системы для международного применения.

Решение SAP Global Trade Services (GTS) представляет собой наиболее зрелый и функционально насыщенный продукт из рассмотренных аналогов. Его основные преимущества - полная интеграция с ERP-системами, поддержка международных стандартов и мощные аналитические возможности. Однако высокая стоимость внедрения и сложность адаптации делают эту систему доступной преимущественно для крупных транснациональных корпораций.

Автоматизированная Информационная Система Таможни (АИСТ) Российской Федерации выделяется развитой интеграцией с системами ЕАЭС и комплексным подходом к управлению таможенными рисками. При этом система испытывает трудности с масштабированием и оперативным обновлением нормативной базы, что снижает ее гибкость в условиях быстро меняющегося законодательства.

Сравнительный анализ позволил выявить, что современные системы электронного декларирования развиваются в следующих ключевых направлениях:

– углубление интеграционных возможностей с международными информационными системами;

– внедрение интеллектуальных алгоритмов анализа рисков;

– развитие мобильных интерфейсов для участников ВЭД;

– повышение устойчивости к пиковым нагрузкам;

– совершенствование механизмов прогнозной аналитики.

На основании проведенного исследования можно сделать вывод о необходимости создания гибридного решения, сочетающего лучшие практики рассмотренных систем - интеграционные возможности SAP GTS, нормативную адаптивность НАСЭД, аналитическую мощь VED1 и межгосударственную совместимость АИСТ. Особое внимание должно быть уделено вопросам масштабируемости и отказоустойчивости проектируемой системы.

**1.4 Постановка задачи**

Разработка любого программного средства начинается с постановки задач, которые должны быть решены в ходе работы, для того чтобы получить конечный продукт. В ходе дипломного проектирования необходимо разработать библиотеку «*CustomsDoc*» для автоматизации ведения таможенных документов. Для этого необходимо решить следующие задачи:

1 Проанализировать исходные данные.

2 Произвести анализ существующих информационных систем.

3 Выбрать язык программирования, средства разработки и сторонние библиотеки. Привести описание выбранного языка программирования, средств разработки и сторонних библиотек. Обосновать выбор языка программирования, средств разработки и сторонних библиотек.

4 Спроектировать структуру информационной системы.

5 Описать и реализовать используемые в системе алгоритмы.

6 Протестировать программное средство.

7 Произвести оценку временных показателей программного средства.

8 Произвести оценку ресурсных показателей программного средства.

9 Ввести в эксплуатацию и обосновать минимальные технические требования к оборудованию.

10 Разработать руководство по эксплуатации программным средством.

11 Произвести технико-экономическое обоснование разработки библиотеки «*CustomsDoc*» для автоматизации ведения таможенных документов.

В рамках дипломного проекта рассматривается процесс ведения таможенных документов, включающий:

– оформление деклараций;

– сохранение таможенных документов;

– расчет таможенных платежей;

– формирование отчетности для государственных и внутренних нужд.

**2 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ**

**2.1 Анализ информационной системы**

Разрабатываемая информационная система предназначена для автоматизации процессов ведения таможенных документов. Ее основное назначение – сокращение временных затрат на оформление деклараций, минимизация ошибок при вводе данных, обеспечение прозрачности документооборота и интеграция с государственными и корпоративными информационными системами. ИС позволит централизованно управлять всеми этапами таможенного оформления.

Основные функции системы включают автоматизированное заполнение таможенных деклараций, проверку корректности данных, расчет таможенных платежей, формирование отчетности и электронный обмен данными с внешними системами. Решаемые задачи охватывают не только непосредственное оформление документов, но и контроль их статусов, уведомление пользователей о изменениях, а также архивирование завершенных операций. Система будет работать в двух основных режимах: интерактивном (ручной ввод и редактирование данных) и автоматическом (фоновые проверки, интеграционные процессы с другими системами).

На рисунке 2.1 показана контекстная диаграмма (*IDEF*0). *IDEF*0 акцентирует внимание на соподчиненности объектов и помогает описать бизнес-процессы. Основная цель *IDEF*0 — представить функциональную модель системы в виде графической схемы. Она позволяет выделить входные и выходные данные, установить механизм и способы управления.

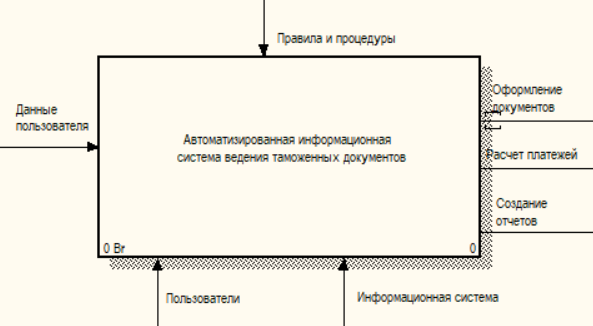


Рисунок 2.1 – Контекстная диаграмма

Входной информацией для ИС служат данные от пользователей (наименования, коды ТН ВЭД, стоимость, вес, страна происхождения и т.д.), сведения о контрактах и поставщиках. Выходная информация включает оформленные таможенные декларации, расчеты платежей, а также аналитические отчеты. Способами управления являются правила и процедуры таможенных органов. Основными получателями выходных данных являются таможенные брокеры, финансовые службы и руководство компании.

Более подробной диаграммой является диаграмма декомпозиции, которая показана на рисунке 2.2. Эта диаграмма является более подробным описание контекстной диаграммы. Графическое представление помогает наглядно описать логику и взаимодействие процессов организации. Входными данными являются данные пользователя, которые приходят в блок «Аутентификации и авторизации», далее переходят в блок «Просмотр требований к документу», потом в «Создание документа», потом в «Проверка форматно-логического контроля» и далее в «Сохранение документа и отправка в таможенные документы». Выходными данными являются создание отчетов и расчет платежей. Все блоки имеет в качестве способов управления правила и процедуры. А в качестве механизмов – пользователи, таможенные органы и информационная система.

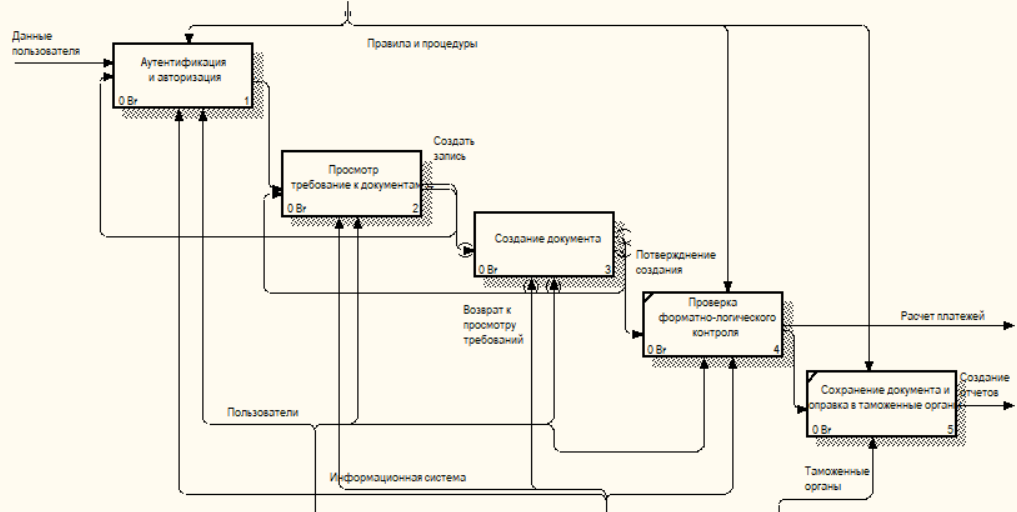


Рисунок 2.2 – Диаграмма декомпозиции

Периодичность решения задач в системе варьируется в зависимости от их типа. Оперативные задачи, такие как подача деклараций и проверка статусов, выполняются в реальном времени или ежечасно. Фоновые процессы, включая синхронизацию с внешними системами и формирование отчетности, могут запускаться ежедневно или еженедельно. Архивация данных и комплексные проверки выполняются ежемесячно.

**2.2 Структура информационной системы**

Разрабатываемая информационная система для ведения таможенных документов состоит из нескольких взаимосвязанных подсистем, каждая из которых отвечает за определенный этап обработки данных и взаимодействия с пользователями. Основу системы составляет модульная архитектура, обеспечивающая гибкость, масштабируемость и удобство сопровождения.

На рисунке 2.3 показана структура ИС. Она состоит из модулей: «Аутентификации и авторизации пользователей», «Регистрации документа», «Форматно-логического контроля», «Модуль создания отчетов».

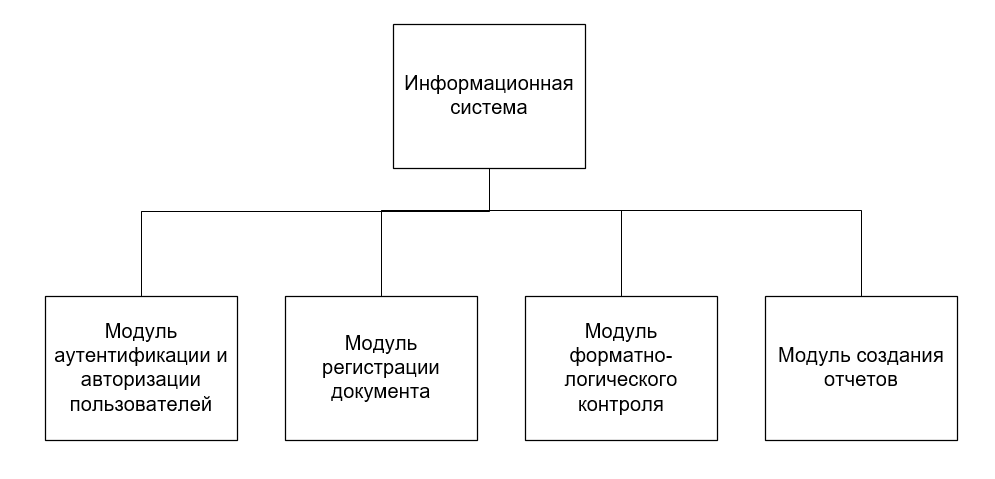


Рисунок 2.3 – Структура системы

Модуль авторизации и аутентификации является ключевым компонентом системы ведения таможенных документов, обеспечивающим безопасный доступ пользователей к функционалу ИС в соответствии с их правами и должностными обязанностями. Он решает задачи подтверждения личности сотрудников, разграничения доступа к данным и операциям, а также защиты системы от несанкционированного вмешательства.

Модуль регистрации документов играет ключевую роль в системе ведения таможенных документов, обеспечивая систематизированный учет и обработку всей входящей и исходящей документации. Его основная задача – придание документам юридической значимости, контроль их жизненного цикла и обеспечение прозрачности документооборота. Благодаря этому модулю каждый документ получает официальный статус в системе, что исключает возможность его потери или несанкционированного изменения. Структура модуля регистрации документа показана на рисунке 2.4.



Рисунок 2.4 – Структура модуля регистрации

Документ поступает в формате *JSON*. После загрузки документу присваивается уникальный идентификатор, и он поступает в очередь на валидацию. На этапе валидации система проверяет корректность и полноту данных документа, сверяя его с установленными требованиями. Автоматически анализируются обязательные поля: номер, дата, наименование груза, код ТН ВЭД, стоимость, реквизиты отправителя и получателя. Если обнаруживаются ошибки (например, несоответствие кода товара классификатору или отсутствие подписи), система помечает документ как «Требует исправления» и уведомляет пользователя об этом. Для снижения количества ошибок применяются предустановленные шаблоны и подсказки. После успешной проверки документ переходит в статус «Проверено» и становится доступным для дальнейшего согласования.

Процесс согласования зависит от типа документа и внутренних регламентов. Система автоматически направляет документ ответственному лицу для согласования документа. На каждом этапе ответственное лицо получает задачу в личном кабинете, а документ временно блокируется для редактирования. Если требуется доработка, пользователь оставляет комментарий, и документ возвращается на предыдущий этап. После финального согласования документ переходит в статус «Утвержден».

Модуль формирования отчетов и документов представляет собой ключевой компонент системы таможенного документооборота, обеспечивающий автоматизированное создание, оформление и выгрузку различных видов отчетной документации. Этот модуль интегрирован со всеми основными подсистемами и предназначен для генерации как регламентированных таможенных форм, так и внутренних аналитических отчетов предприятия. Основные функции модуля включают автоматизированное заполнение стандартных форм таможенных документов, создание пользовательских отчетов, экспорт данных в требуемых форматах и визуализацию статистической информации.

Информационная система базируется на концепции разделения приложения на клиентскую и серверную части. Схема разделения приложения показана на рисунке 2.5.

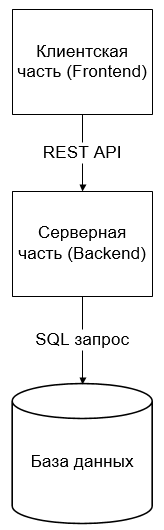


Рисунок 2.5 – Схема разделения приложения

Frontend-разработка включает в себя создание пользовательских интерфейсов веб-приложений. Это то, что видит и с чем взаимодействует пользователь. Важным аспектом фронтенда является работа с API для получения и отправки данных на сервер.

Backend-разработка отвечает за логику приложения и взаимодействие с базой данных. Она включает в себя серверную часть, которая обрабатывает запросы от клиента, управляет данными и выполняет бизнес-логику. Серверная часть также включает в себя создание RESTful API, которые позволяют фронтенду и бекенду взаимодействовать друг с другом.

REST (Representational State Transfer) — это архитектурный стиль, который позволяет различным компонентам приложения обмениваться данными через HTTP-запросы. Взаимодействие между фронтендом и бекендом происходит с использованием методов, таких как GET, POST, PUT и DELETE. Фронтенд отправляет запросы к API, а бекенд обрабатывает их, выполняя необходимые операции с данными и возвращая ответ в формате JSON или XML.

База данных хранит все данные приложения и обеспечивает их структурированный доступ. Бекенд взаимодействует с базой данных через ORM запросы, что позволяет управлять данными, обеспечивая их целостность и безопасность. ORM (Object-Relational Mapping) — это технология, позволяющая взаимодействовать с реляционными базами данных, используя объектно-ориентированные подходы. Она служит в качестве промежуточного слоя между объектами в коде и таблицами в базе данных, что упрощает процесс работы с данными. Вместо написания сложных SQL-запросов возможно использование объектных моделей. ORM преобразует объекты в записи базы данных и наоборот. Когда разработчик создает объект в коде, ORM автоматически генерирует соответствующий SQL-запрос для сохранения этого объекта в базе. При извлечении данных из базы ORM преобразует записи обратно в объекты, что упрощает дальнейшую работу с ними. Это позволяет сосредоточиться на логике приложения, не углубляясь в детали реализации базы данных.

**2.3 Алгоритмическое обеспечение**

**2.3.1 Алгоритм регистрации документа**

Процесс регистрации документа начинается с получения данных. Система анализирует тип документа (товарная декларация ДТ или транспортный документ ТС) и направляет его в соответствующий обработчик. Для товарных деклараций система выполняет комплексное сохранение данных: проверяет информацию о декларанте (при необходимости создает новую запись), сохраняет все товарные позиции с их характеристиками и прикрепленные файлы. Для транспортных документов сохраняются технические параметры, стоимость и год выпуска транспортного средства. На этом этапе генерируется уникальный код документа и устанавливается начальный статус "Создан". После успешного сохранения документа запускается процесс формально-логического контроля (ФЛК). Система выполняет набор проверок, определенных для данного типа документа. Результат проверки фиксируется в системе: при успешном прохождении документ получает статус "ФЛК не пройден", при обнаружении несоответствий - "ФЛК пройден". Особенностью данного этапа является дифференцированный подход – для разных типов документов применяются различные наборы проверок.

Для документов, успешно прошедших ФЛК, автоматически рассчитываются таможенные платежи. Для товарных деклараций расчет основан на кодах ТН ВЭД с учетом количества, веса и стоимости товаров. Для транспортных средств учитывается возраст, стоимость и объем двигателя. Результаты расчета сохраняются в системе и становятся частью документа.

На завершающем этапе документ получает финальный статус: "На рассмотрении" для успешно обработанных или "Ошибка ФЛК" для документов с выявленными несоответствиями. Все действия системы логируются, что обеспечивает прозрачность процесса и возможность последующего аудита. Пользователь получает соответствующее уведомление о результате обработки документа. Схема алгоритма показана на рисунке 2.6

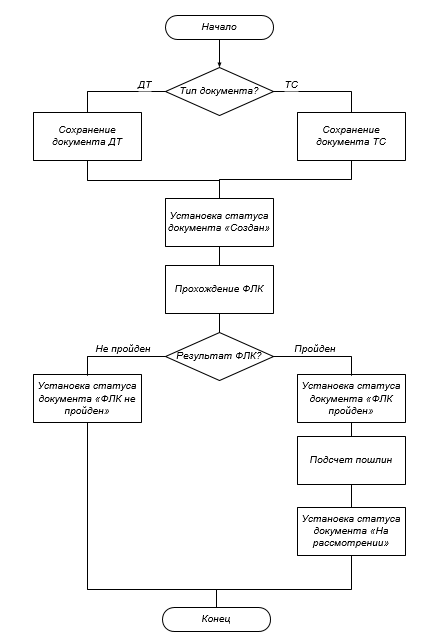


Рисунок 2.6 – Схема алгоритма регистрации документа

**2.3.2 Алгоритм расчета пошлин**

При сохранении документов необходим расчет пошлин для товаров и транспортных средств. Для товаров не облагающимися акцизами вес не должен превышать 30 килограмм или стоимость не должна превышать 10000 евро. Для товаров облагающимися акцизами добавляются дополнительные пошлины. Пошлина транспортных средств вычитывается исходя из возраста транспорта, заявленной стоимости и объема двигателя. После выполнения всех расчетов система автоматически сохраняет полученные значения пошлин в соответствующих записях базы данных. Каждая операция сопровождается записью в системный журнал с указанием кода документа и временной меткой. Такой подход обеспечивает прозрачность расчетов и возможность последующего аудита. Схема алгоритма показана на рисунке 2.7.

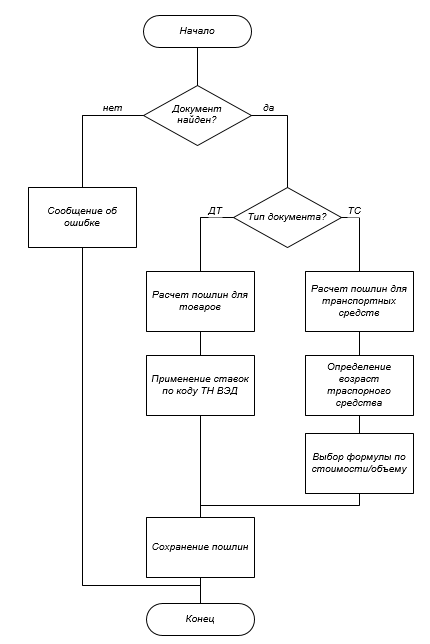


Рисунок 2.7 – Схема алгоритма подсчета пошлин

**2.4 Информационное обеспечение**

Система оперирует тремя ключевыми категориями данных: входными (данные пользователей), хранимыми (база данных и справочники) и выходными (отчеты, документы). Входные данные информационной системы показаны в таблице 2.1.

Таблица 2.1 Входные данные системы

|  |  |  |
| --- | --- | --- |
| Этап ввода | Тип ввода | Входные данные |
| Процесс регистрация пользователя | Ручной ввод | Имя пользователя |
| Пароль |
| Email |
| Процесс авторизации | Ручной ввод | Имя пользователя |
| Пароль |
| Процесс создания документа | Ручной ввод | ФИО декларанта |
| Паспортные данные |
| Адрес проживания |
| Наименование товара |
| Вес товара |
| Количество товаров |
| Код ТН ВЕД товара |
| Цена товара |
| Марка и модель ТС |
| Цена ТС |
| Объем двигателя ТС |
| Вин код ТС |
| Год выпуска |
| Цвет ТС |
| Страна отправитель |
| Страна получатель |
| Дополнительные файлы |
| Причина |

**2.5 База данных**

База данных – это организованная коллекция данных, которая хранится и обрабатывается с помощью компьютерных систем. Базы данных используются для хранения, поиска, изменения и анализа больших объемов информации, такой как персональные данные, финансовые транзакции, научные публикации и т.д.

Реляционная база данных – это набор данных с предопределёнными связями между ними. Эти данные организованы в виде набора таблиц. Состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определённый тип данных, в каждой ячейке – значение атрибута. Каждая стока таблицы представляет собой набор связанных значений, относящихся к одному субъекту или сущности.

Можно выделить основные требования к базе данных информационной системы:

– надёжность, обеспечивающая возможность непрерывной корректной работы;

– защищенность, гарантирующая соблюдение правил обработки информации, конфиденциальность;

– полнота, позволяющая пользователям иметь все необходимые данные;

– нормализация данных для исключения дублирования и аномалий.

Нормализация в базах данных является процессом организации структуры данных в соответствии с набором правил, называемых нормальными формами. Цель нормализации состоит в устранении избыточности данных, обеспечении целостности и минимизации аномалий при обновлении, вставке и удалении данных. Требуется соблюсти соответствие базы данных трём нормальным формам.

Первым шагом в проектировании базы данных является определение ее структуры и основных сущностей. Каждая таблица имеет свои поля, которые содержат информацию о соответствующей сущности. Установлены связи между таблицами, чтобы обеспечить целостность данных и возможность получения связанной информации.

Для управления базой данных использовано специальное программное обеспечение, такое как система управления базами данных *PostgeSQL*. Оно позволяет создавать таблицы, добавлять, изменять и удалять записи, а также выполнять запросы для получения нужной информации. Обеспечена безопасность данных и доступа к базе данных для разных пользователей.

На рисунке 2.5 показана ER-диаграмма в нотации Чена для модуля регистрации документа. Основными элементами нотации являются сущности, связи и атрибуты. Сущности, изображаемые прямоугольниками, представляют ключевые объекты предметной области - такие как пользователи, документы или товары. Связи между сущностями, обозначаемые ромбами, показывают характер взаимодействия между объектами. Особое значение в нотации Чена уделяется кардинальности связей, которая точно определяет количественные соотношения между связанными сущностями. Атрибуты в нотации Чена изображаются овалами и соединяются с соответствующими сущностями или связями.

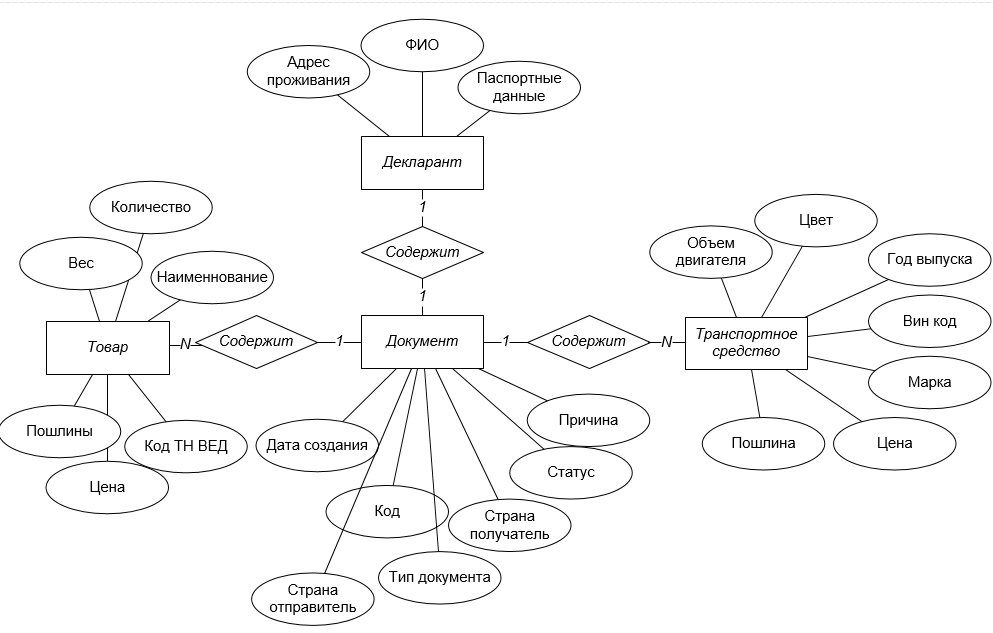


Рисунок 2.5 – Нотация Чена для модуля регистрации документов

Таблица *flk*\_*checks* необходима для хранения проверок, которые будет проходить документ в форматно-логическом контроле. Для каждого типа документа существуют свои проверки. Поля этой таблицы показаны в таблице 2.1.

Таблица 2.1 Структура таблицы *flk*\_*checks*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *code* | *varchar(16)* | Код ошибки |
| *description\_info* | *varchar(255)* | Информация по проверке |
| *description\_error* | *varchar(255)* | Сообщение, выводимое при ошибке |
| *isActive* | *boolean* | Активность проверки |
| *document\_type* | *timestamp* | Тип документа, для которого написана проверка |
| *node\_name* | *varchar(32)* | Наименование поля ошибки |
| *node\_number* | *integer* | Номер поля ошибки |
| *date\_start* | *timestamp* | Дата начала |
| *date\_end* | *timestamp* | Дата конца |

В таблице *common\_reports* хранятся записи о прохождении документа форматно-логического контроля. У этой таблицы есть связь с таблицей *flk*\_*checks* и при необходимости можно узнать все ошибки документа. Поля этой таблицы показаны в таблице 2.2.

Таблица 2.2 Структура таблицы *common\_reports*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *document\_code* | *varchar(32)* | Код документа |
| *description* | *varchar(255)* | Описание проверки документа |
| *date\_check* | *timestamp* | Время проверки документа |

Таблица *types\_document* необходима для хранения всех документов, которые обрабатывает информационная система. Поля этой таблицы показаны в таблице 2.3.

Таблица 2.3 Структура таблицы *types\_document*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *name* | *varchar(32)* | Название документа |
| *version* | *varchar(32)* | Версия документа |
| *isActive* | *boolean* | Активность документа |

Таблица *documents* необходима для хранения документов типа ДТ и ТС. Поля этой таблицы показаны в таблице 2.4.

Таблица 2.4 Структура таблицы *documents*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *declarant\_id* | *integer* | Идентификатор записи декларанта |
| *document\_code* | *varchar(32)* | Код документа |
| *sender* | *varchar(50)* | Страна отправления |
| *recipient* | *varchar(50)* | Страна получения |
| *reason* | *varchar(100)* | Причина |
| *status* | *varchar(32)* | Статус документа |
| *inspector\_id* | *integer* | Идентификатор записи таможенного лица, принимающее документ |
| *user\_id* | *integer* | Идентификатор пользователя |
| *signature* | *text* | Электронная подпись |
| *type\_document* | *varchar(32)* | Тип документа |
| *special\_mark* | *text* | Заметки |
| *date\_created* | *timestamp* | Дата создания |
| *date\_updated* | *timestamp* | Дата обновления |

Таблица *goods* хранит в себе все товары, которые описываются в документе ДТ. Поля этой таблицы показаны в таблице 2.5.

Таблица 2.5 Структура таблицы *goods*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *document\_id* | *integer* | Идентификатор записи документа |
| *name* | *varchar(100)* | Имя товара |
| *codeEA* | *varchar(32)* | Код ТН ВЕД |
| *amount* | *integer* | Количество |
| *weight* | *float* | Вес |
| *cost* | *float* | Цена |
| *currency* | *varchar(5)* | Валюта |
| *tax* | *double* | Пошлина |

Таблица *vehicles* хранит в себе все транспортные средства, которые описываются в документе ТС. Поля этой таблицы показаны в таблице 2.5.

Таблица 2.5 Структура таблицы *vehicles*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *document\_id* | *integer* | Идентификатор записи документа |
| *brand* | *varchar(64)* | Марка и модель ТС |
| *vin\_code* | *varchar(32)* | Вин код |
| *release\_year* | *integer* | Год выпуска |
| *color* | *varchar(16)* | Цвет |
| *engine\_capacity* | *integer* | Объем двигателя |
| *cost* | *integer* | Цена |
| *currency* | *varchar(8)* | Валюта |
| *tax* | *double* | Пошлина |

Таблица *declarant* хранит в себе информацию о декларантах, подающих документы. Поля этой таблицы показаны в таблице 2.6.

Таблица 2.6 Структура таблицы *declarant*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *fio* | *varchar(100)* | ФИО декларанта |
| *passport* | *varchar(50)* | Паспортные данные декларанта |
| *address* | *varchar(100)* | Адрес регистрации декларанта |

Таблица *codes\_tnved* является справочником для хранения кодов ТН ВЕД. Поля этой таблицы показаны в таблице 2.5.

Таблица 2.6 Структура таблицы *codes\_tnved*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *code* | *integer* | Код ТНВЕД |
| *description* | *varchar(256)* | Описание товара |

Таблица *files\_document* необходима для хранения файлов, прилагаемых к документам. Поля этой таблицы показаны в таблице 2.5.

Таблица 2.6 Структура таблицы *files\_document*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *document\_id* | *integer* | Идентификатор записи документа |
| *file\_data* | *bytea* | Данные файла в виде байтов |
| *file\_name* | *varchar(256)* | Название файла |

Таблица *messages\_log* хранит информацию о всех действиях, проводимых в информационной системе. Поля этой таблицы показаны в таблице 2.5.

Таблица 2.6 Структура таблицы *messages\_log*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *message* | *text* | Сообщение |
| *user\_id* | *integer* | Идентификатор пользователя, производивший действие |
| *document\_code* | *varchar(32)* | Код документа |
| *date* | *timestamp* | Время произведения действия |

Таблица users хранит информацию о пользователях. Поля этой таблицы показаны в таблице 2.5.

Таблица 2.6 Структура таблицы *users*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *username* | *varchar(32)* | Логин пользователя |
| *password* | *varchar(256)* | Пароль |
| *email* | *varchar(64)* | Электронная почта |

Таблица roles хранит информацию о пользователях. Поля этой таблицы показаны в таблице 2.5.

Таблица 2.6 Структура таблицы *roles*

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| *id* | *integer* | Идентификатор записи |
| *name* | *varchar(32)* | Имя роли |

Все связи между таблицами описаны в таблице .

Таблица 2. Описание связей между таблицами

|  |  |  |
| --- | --- | --- |
| Название таблицы 1 | Название таблицы 2 | Описание |
| Пользователи | Роли | У одного пользователя может быть несколько ролей |
| Ошибки ФЛК | Отчеты Проверок | Одному отчету соответствуют несколько ролей |
| Ошибки ФЛК | Тип документа | У каждого типа документа есть набор ошибок ФЛК |
| Тип документа | Документы | Каждому документу соответствует тип документа |
| Документы | Товары | У одного документа может быть несколько товаров |
| Документы | Транспортные средства | У одного документа может быть несколько ТС |
| Документы | Файлы документа | У одного документа может быть несколько файлов |
| Документы | Декларанты | У одного документа есть один декларант |

**2.6 Техническое программное обеспечение**

Аппаратные требования включают серверное оборудование на базе многоядерных процессоров Intel Xeon или AMD EPYC с тактовой частотой не менее 2.5 ГГц. Минимальный объем оперативной памяти должен составлять 32 ГБ для рабочих серверов. Система хранения должна быть организована на SSD-накопителях с общей емкостью не менее 1 ТБ для операционных данных.

Сетевое оборудование должно обеспечивать высокоскоростное соединение не менее 1 Гбит/с между компонентами системы.

В качестве операционной системы выбран Linux. Этот выбор обусловлен рядом технических и экономических преимуществ, делающих Linux оптимальной платформой для развертывания критически важных приложений. С точки зрения производительности Linux демонстрирует оптимальные характеристики для работы с высокими нагрузками. Среди доступных дистрибутивов Linux для проекта выбран Ubuntu Server. Этот вариант обеспечивает долгосрочную поддержку в течение пяти лет, обладает обширной экосистемой готовых пакетов и отличается простотой настройки. Наличие официальных репозиториев для Docker, Kubernetes и PostgreSQL значительно упрощает процесс развертывания и обслуживания инфраструктуры.

Для обеспечения переносимости и согласованности окружений в проектируемой информационной системе используется технология контейнеризации Docker. Каждый функциональный модуль системы (веб-приложение, сервис обработки данных, СУБД) развертывается в виде изолированного Docker-контейнера с точно определенными зависимостями и параметрами окружения. Такой подход позволяет:

– гарантировать идентичность сред разработки и тестирования;

– минимизировать конфликты версий программных зависимостей;

– упростить процесс развертывания и обновления компонентов;

– оптимизировать использование ресурсов серверной инфраструктуры.

Для управления кластером Docker-контейнеров в промышленной эксплуатации применяется платформа Kubernetes, которая обеспечивает:

– автоматическое масштабирование сервисов в зависимости от нагрузки;

– отказоустойчивость за счет самовосстановления контейнеров;

– балансировку нагрузки между экземплярами сервисов;

– управление сетевым взаимодействием компонентов;

– контроль потребления ресурсов каждым контейнером.

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Чудинов, И.Л. Информационные системы и технологии / И. Л. Чудинов, В. В. Осипов – Томск : Питер ком, 2013. – 145 с.

[2] Боггс, У. *UML* и Rational Rose / У. Боггс, М. Боггс. – М. : ДМК, 2002. – 510 с.

[3] Чудинов, И.Л. Информационные системы и технологии / И. Л. Чудинов, В. В. Осипов – Томск : Питер ком, 2013. – 145 с.

[4] Вендров, А. М. *CASE*–технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М. : Финансы и статистика, 1998. – 275 с.

[5] Нормализация отношений. Шесть нормальных форм [Электронный ресурс]. – Режим доступа: *https://habr.com/post/254773/.*

[6] *PHP*. Документация [Электронный ресурс]. – Режим доступа: *https://www.php-info.ru/.*