Техпроект на разработку системы управления интернет-магазином автозапчастей.

[Цель документа](#_ta8uep5wsohn)

[Сценарии использования](#_qcgjjgiljfx7)

[Оператор 1С, менеджер сайта](#_mt570x77f5ou)

[Сценарий 1 - Работа с контрагентами](#_v693c2bxzipc)

[Сценарий 2 - Работа с группами скидок](#_f1v0ussaf5gn)

[Сценарий 3 - Работа с прайсами](#_a2zh2hi4idln)

[Сценарий 4 - Работа с документами](#_ykjnfd1iqg87)

[Сценарий 5 - Обработка заказов](#_z1955vdu6mr5)

[Сценарий 6 - Обработка заказанных позиций](#_a2mm884faank)

[Сценарий 7 - Общение с клиентом по заказу](#_r5pazn4fc8te)

[Сценарий 8 - Редактирование перечня тем для сообщений](#_ehjwyq520cx7)

[Сценарий 9 - Редактирование шаблонов уведомлений (email, sms)](#_5vvc2i4en8nv)

[Пользователь сайта](#_kb9r0elh55cg)

[Сценарий 1 - Поиск запчастей](#_69ii7ofrnco1)

[Сценарий 2 - Регистрация](#_it3xracg6rnw)

[Сценарий 3 - Оформление заказа](#_1h7dgmdeywml)

[Сценарий 4 - Личный кабинет](#_5caluax5beix)

[Требования к системе](#_romccvz8j0y9)

[Синхронизация справочников и документов системы 1С и СУИМ](#_f1duq3sejbqj)

[Создание и управление заказами пользователей](#_5m53wjpmtb92)

[Создание и управление заказами поставщикам](#_xki2ple188i5)

[Обеспечение выверенной экосистемы во взаимодействии с контрагентами](#_4528sufloh0v)

[Обеспечение гибкой системы ценообразования;](#_ohfyfsf0jps2)

[Создание базы прайсов с запчастями из различных источников с возможностью осуществления выборки](#_1hggu6nglmeo)

[Управление контентом сайта для удобного наполнения](#_z22qmpuhf562)

[Создание единого интерфейса для интеграции с системой.](#_s88dbcqcoz6x)

[Архитектура](#_b088r8qrjiik)

[Реализация](#_plxebhs7hezw)

[Интеграция с 1С](#_h28tvexdwro2)

[Механизмы взаимодействия](#_puki8mh94gzq)

[REST интерфейс](#_478p44pxx3tf)

[Приложение-потребитель COM](#_u7p9a318olli)

[Синхронизация данных](#_a2mzxj7hfzae)

[Формирование сообщения обмена](#_im1e3jgqdzya)

[Уведомление о получении изменений](#_vnkdqcv2rwgw)

[Хранилище](#_br1fri32egno)

[Back-End (серверная часть)](#_l1itnwxqfrqf)

[Административная панель](#_k9ozpd7lodue)

[Front-End (веб-сайт)](#_7lfv971k7bj0)

[Миграция данных с системы PartsSoft](#_gf2ij9ko9nj)

[Документация](#_cv26ke910nuw)

[Руководство администратора](#_khumrqydskye)

[Руководство разработчика](#_o5851090ghla)

# Цель документа

Выделение приоритетных сценариев использования, формирование требований к решению, составление модели решения и детального описания реализации.

# Сценарии использования

*В разработке программного обеспечения и системном проектировании это описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды.*

## Оператор 1С, менеджер сайта

В лице актантов выступают сотрудники магазина с целью повышения эффективности и автоматизации работы. Предполагается, что работа осуществляется пользователем либо из административной панели в окне браузера, либо в системе 1С:Предприятие любой конфигурации.

### Сценарий 1 - Работа с контрагентами

Пользователь переходит в **раздел управления контрагентами**.

Пользователь может создать нового контрагента или редактировать существующих (профиль, ценообразование).

Пользователь может видеть движение баланса, документы (счета, накладные, заказы) того или иного контрагента.

Пользователь может уведомить клиента одним из средств связи.

Пользователь может инициировать оформление нового заказа.

### Сценарий 2 - Работа с группами скидок

Пользователь может создавать новые группы скидок.

Пользователь может устанавливать группу по умолчанию для новых контрагентов, а также группу для незарегистрированных пользователей.

### Сценарий 3 - Работа с прайсами

Пользователь может добавлять прайсы на основе контрагента с типом **Поставщик**, которые будут задействованы в проценке.

Пользователь может выбрать источник позиций для прайса (файл, веб-сервис, email)

Пользователь может настроить наиболее важные позиции из прайсов, которые будут выводиться в приоритетном порядке при проценке.

Пользователь может активировать или отключить прайс лист, чтобы убрать его из проценки.

Пользователь может посмотреть статистику заказов по выбранному прайсу.

### Сценарий 4 - Работа с документами

Пользователь может просматривать и отправлять на печать автогенерируемые документы.

### Сценарий 5 - Обработка заказов

Пользователь видит новый заказ (1C / Админ-панель)

После проверки информации переводит заказ в обработку.

Переходит к обработке заказанных позиций.

### Сценарий 6 - Обработка заказанных позиций

Пользователь видит заказанную позицию.

Пользователь переводит состояние заказанной позиции из Принят к обработке в состояние В работе

Если заказ был осуществлен у внешнего поставщика, пользователь видит позицию в заказе поставщику.

В зависимости от типа прайса поставщика (веб-сервис, файл) пользователь обновляет статусы позиций вручную, либо это происходит автоматически.

### Сценарий 7 - Общение с клиентом по заказу

Пользователь переходит в раздел **Сообщения**, видит список активных диалогов по заказам.

При необходимости осуществляет фильтрацию диалогов по контрагенту, теме, дате.

Выбирает необходимый диалог, переходит в окно чата.

Пользователь имеет возможность прикрепить файлы к беседе.

При наличии непрочитанных сообщений пользователь видит соответствующий индикатор.

### Сценарий 8 - Редактирование перечня тем для сообщений

Пользователь переходит в раздел **Сообщения**, нажимает на кнопку **Редактировать темы**.

Осуществляет редактирование.

### Сценарий 9 - Редактирование шаблонов уведомлений (email, sms)

Пользователь переходит в **Настройки шаблонов**.

Пользователь может создавать или редактировать существующие шаблоны.

## Пользователь сайта

Любая категория пользователей.

### Сценарий 1 - Поиск запчастей

Пользователь осуществляет поиск необходимой детали с помощью единого поиска на сайте.

В зависимости от выбранных параметров поиска происходит поиск по каталогу запчастей на основных складах, либо показываются результаты проценки.

Пользователь видит цены в зависимости от активной группы скидок.

Пользователь может перейти в карточку запчасти, в которой отображаются характеристики и перечень аналогов.

Пользователь может осуществить поиск аналогов или начать проценку непосредственно на основе результатов поиска.

Пользователь может добавить позицию в корзину в необходимом количестве.

### Сценарий 2 - Регистрация

Пользователь выбирает Физическое лицо или Юридическое лицо.

Заполняет необходимые поля.

Переходит в контекст сайта для зарегистрированных пользователей.

Имеет возможность зайти в личный кабинет.

### Сценарий 3 - Оформление заказа

Пользователь добавляет необходимые позиции в корзину.

Пользователь переходит в корзину и инициирует процесс оформления заказа.

Если пользователь не зарегистрирован, начинается процесс регистрации.

После успешной регистрации пользователь продолжает оформление заказа.

Пользователь может видеть статус заказа и начать диалог с менеджером в рамках того или иного заказа.

### Сценарий 4 - Личный кабинет

В личном кабинете пользователь имеет возможность наблюдать за выполнением заказа, просматривать документы, отправлять и получать сообщения от менеджера, видеть свой баланс.

# Требования к системе

Программный комплекс предназначен для решения следующих задач:

* Синхронизация справочников и документов системы 1С и хранилища системы управления интернет-магазином (далее СУИМ);
* Создание и управление заказами клиентов;
* Создание и управление заказами поставщикам;
* Обеспечение выверенной экосистемы во взаимодействии с контрагентами;
* Обеспечение гибкой системы ценообразования;
* Создание базы прайсов с запчастями из различных источников с возможностью осуществления выборки;
* Управление контентом сайта для удобного наполнения;
* Создание единого интерфейса для интеграции с системой.

## Синхронизация справочников и документов системы 1С и СУИМ

В рамках данной задачи необходимо наличие слоя для обеспечения запросов и передачи данных между системами.

В хранилище СУИМ должны загружаться следующие справочники:

* Номенклатура;
* Авторазбор.

Документы:

* Накладные;
* Счета.

В систему 1C должны отправляться:

* Контрагенты;
* Заказы;

При изменении документов новая информация должна попадать в СУИМ.

## Создание и управление заказами пользователей

Заказ может быть создан либо через сайт, либо вручную в административной панели (далее АП).

При создании добавляются необходимые позиции с использованием механизма поиска из прайсов (проценки) или по каталогу.

Формируется заказ, который состоит из одной или нескольких позиций. Каждая заказанная позиция является отдельной сущностью и имеет свой жизненный цикл. Заказ также имеет жизненный цикл.

После создания заказа в СУИМ:

* Автоматически создается документ **Заказ клиента** в системе 1C;
* Автоматически создается документ **Счет на оплату** в системе 1C;
* **Счет на оплату** попадает в СУИМ с целью мониторинга, отслеживания статуса и выставления клиенту для оплаты;

Добавленные к заказу позиции на сайте также добавляются к заказу в 1С по следующей схеме: если номенклатура присутствует в базе, то добавляем существующую позицию к заказу, иначе создается новая запись в отдельной папке, изолированной от синхронизации с хранилищем СУИМ.

При обновлении статуса заказа в 1С также обновляется и статус заказа в СУИМ.

После реализации заказа формируется документ **Накладная** и также отправляется в СУИМ.

## Создание и управление заказами поставщикам

Если заказанная позиция находится у поставщика, она автоматически попадает в заказ поставщику.

При наличии у поставщика веб-сервиса заказ может быть сформирован и оформлен автоматически с использованием специального API. По возможности необходимо автоматическое отслеживание позиций, заказанных у поставщиков.

В дополнение, необходимо предусмотреть возможность автоматического добавления к заказу поставщика позиций с остатком ниже установленного минимума для поддержки наличия товара.

## Обеспечение выверенной экосистемы во взаимодействии с контрагентами

Пользователи сайта должны иметь возможность отслеживать состояния своих заказов, задавать вопросы менеджеру и получать необходимые документы. Клиент может быть уведомлен в любое время одним из способом связи.

## Обеспечение гибкой системы ценообразования;

Для каждого контрагента необходимо реализовать возможность указывать группу скидок.

В дополнение необходимо реализовать подобную систему для редактирования коэффициентов цен в разрезе складов.

К примеру, запчасти на удаленном складе будут иметь наценку.

В результате цена должна образовываться на основе различных коэффициентов и условий.

## Создание базы прайсов с запчастями из различных источников с возможностью осуществления выборки

Необходимо предусмотреть загрузку прайсов из файла, автоматически из email по ключевым словам, от веб-сервиса поставщика.

Позиции (остатки, цены) из прайсов должны синхронизироваться с 1C параллельно с синхронизацией справочника **Номенклатура**.

Поиск позиций из прайсов (проценка) должна осуществляться по наиболее оптимальным алгоритмам с учетом частичной выдачи результатов.

Необходимо реализовать возможность приоритезации тех или иных позиций прайсов по заданным критериям.

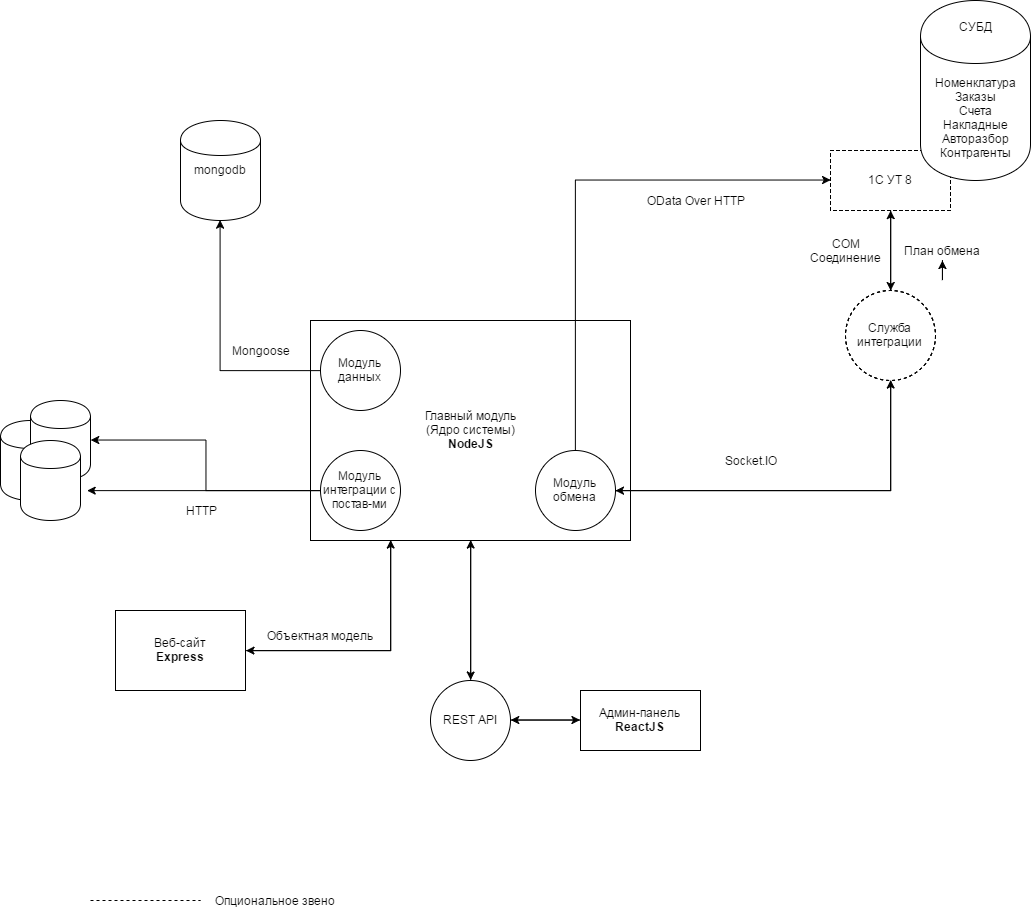
## Управление контентом сайта для удобного наполнения

Система должна следовать канонам систем управления контентом. Необходима возможность редактирования иерархии страниц сайта и добавления контента с использованием WYSIWYG редактора.

## Создание единого интерфейса для интеграции с системой.

Система должны быть абстрагирована от визуальной составляющей и иметь внешний программный интерфейс. Доступ к интерфейсу должен быть безопасным и иметь одну точку входа.

# Архитектура



**https://www.draw.io/#G0BwcePxo3QA\_LeFg2T29EbGVxSjg**

# Реализация

В данном разделе описывается техническая часть.

Более подробная информация по реализации будет описана в документации разработчика.

## Интеграция с 1С

### Механизмы взаимодействия

На данный момент рассматривается 2 механизма взаимодействия.

Текущая версия ядра не позволяет использовать первый более современный и удобный механизм, поэтому решается вопрос обновления ядра до версии 8.3. При невозможности обновления есть вариант вернуться к классическому механизму, для которого будет необходимо разработать отдельное приложение.

#### **REST интерфейс**

Данный механизм позволяет осуществлять взаимодействие с 1С из внешних систем по протоколу OData. 1C автоматически создает веб-сервис и открывает доступ к системе.

Использовать стандартный интерфейс OData прикладного решения просто:

В конфигураторе REST интерфейс публикуется на веб-сервере;

После этого объекты прикладного решения становятся доступны через этот интерфейс;

Способы аутентификации OData клиентов полностью совпадают со способами, используемыми для веб-сервисов;

OData клиенты могут запросить через HTTP документ метаданных, описывающий доступные объекты прикладного решения;

OData клиенты выполняют операции создания, чтения, модификации и удаления данных прикладного решения.

При получении списков данных можно использовать стандартные условия фильтрации ODATA запросов. Например, получить товары, у которых цена меньше или равна 3,5 или больше 200:

**GET /OData\_Tests\_Infobase/odata/standard.odata/Catalog\_Goods?$filter=Price le 3.5 or Price gt 200**

Результат получаем в формате **JSON**.

Основное преимущество данного механизма для нас в том, что нет необходимости создавать отдельное приложение, которое будет реализовывать лишь часть функций и служить дополнительным звеном. Больше звеньев - больше вероятность поломки. При данном подходе мы имеем весь функционал из коробки.

Основная проблема в необходимости обновления версии ядра 1C до 8.3 (текущая 8.2).

#### **Приложение-потребитель COM**

Для организации доступа к данным системы 1С:Предприятие 8 через внешнее соединение, выполняется следующая последовательность действий:

* создается менеджер COM-соединений, с помощью которого производится установка соединения;
* через полученный объект внешнего соединения производится обращение к допустимым методам, свойствам и объектам базы данных, с которой установлено соединение.

Создается Windows приложение, которое затем будет подключаться к нашей серверной части через библиотеку Socket.IO. Приложение находится на той же машине, где расположен сервер 1C. Создается постоянное COM соединение. При данном подходе данные передаются в СУИМ по запросу.

### Синхронизация данных

Для уменьшения избыточности передаваемых данных было решено задействовать технологию планов обмена. Любые изменения справочников, документов в 1C можно автоматически регистрировать. При запросе из СУИМ новых данных 1C вернет только обновленную актуальную информацию и очистит план обмена.

#### **Формирование сообщения обмена**

Для формирования сообщения обмена необходимо выполнить POST-запрос с использованием URL следующего вида: **http://host/base/odata/standard.odata/SelectChanges?<params>**. Где в качестве параметров необходимо указать следующее:

* Параметр DataExchangePoint – должен содержать канонический URL сущности требуемого элемента плана обмена;
* Параметр MessageNo – должен содержать номер сообщения обмена данными,

который будет сформирован в результате данного вызова.

В результате, полный URL для формирования сообщения обмена, будет выглядеть следующим образом: **http://host/base/odata/standard.odata/SelectChanges?DataExchangePoint='http://host/base/odata/standard.odata/ExchangePlan\_ОбменДанными(guid'value')'&MessageNo=34.**

В результате будет получен список изменений, которые необходимо передать в другой узел, в виде потока atom-feed. Каждый элемент будет представлен в формате atom-entry, а удаленные элементы будет представлены в формате atom-deleted-entry (RFC 6721, **http://tools.ietf.org/html/rfc6721**, на английском языке).

#### **Уведомление о получении изменений**

Для уведомления сервера о том, что сообщение обмена успешно получено, необходимо выполнить POST-запрос с использованием URL следующего вида: **http://host/base/odata/standard.odata/ NotifyChangesReceived?<params>**. Где в качестве параметров необходимо указать следующее:

* Параметр DataExchangePoint – должен содержать канонический URL сущности требуемого элемента плана обмена;
* Параметр MessageNo – должен содержать номер сообщения обмена данными, подтверждение получения которого необходимо зафиксировать.

В результате, полный URL для подтверждения получения сообщения обмена, будет выглядеть следующим образом: **http://host/base/odata/standard.odata/NotifyChangesReceived?DataExchangePoint='http://host/base/odata/standard.odata/ExchangePlan\_ОбменДанными(guid'value')'&MessageNo=34.**

## Хранилище

В качестве технологии базы данных будет использована документо-ориентированная СУБД **mongodb** в связке с **Mongoose**.

Схема базы данных:

(TODO). Место для схемы.

## Back-End (серверная часть)

Серверная часть будет разрабатываться на платформе **NodeJS**. Данная платформа хорошо подходит для распараллеливания задач, обладает высокой отказоустойчивостью и прекрасно работает в связке с NOSQL СУБД **mongodb**, которая знаменита своей скоростью обработки запросов.

Для наращивания функционала в будущем будет реализована система событий, на которые могут подписываться вновь созданные модули. Таким образом нам не нужно будет дорабатывать ядро для новых фич. Подробнее об этом будет написано в документации разработчика.

Для взаимодействия с серверной частью будет реализована объектная модель и RESTful API.

## Административная панель

Для комфортной работы менеджеров и отзывчивости был выбран принцип реализации SPA (Single Page Application). В качестве движка работы интерфейса и взаимодействия с серверной частью будет использоваться **ReactJS/Redux**.

## Front-End (веб-сайт)

Для автоматизации построения веб-страниц будет использован система модулей **Express** для **NodeJS** с шаблонизатором **Jade**. В качестве веб-сервера был выбран **nginx**.

# Миграция данных с системы PartsSoft

Все необходимые данные (например база кроссов) будут экспортированы в новую систему.

# Документация

В процессе разработки системы будут разработаны документы:

* Руководство администратора;
* Руководство разработчика.

## Руководство администратора

Будет содержать подробную информацию по всем бизнес-процессам, реализованным в системе.

## Руководство разработчика

Будет содержать информацию по разработке новых модулей без изменения текущего ядра. Также, в него войдут детальные сведения по реализации всех механизмов и аспектов системы.