МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет
«Московский институт электронной техники»

Факультет микроприборов и технической кибернетики
Кафедра информатики и программного обеспечения вычислительных систем

Леонтьев Вадим Вячеславович

Бакалаврская работа
по направлению 09.03.04 «Программная инженерия»

Разработка программного модуля интеграции данных между сайтами для электронной торговли

Студент _____ Леонтьев В.В.

Научный руководитель,
к.п.н., доцент _____ Федотова Е.Л.

Москва 2016

СОДЕРЖАНИЕ

СОКРАЩЕНИЯ....4
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ....5
ВВЕДЕНИЕ....6
1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ....8
1.1 Предварительные исследования....8
1.2 Анализ существующих программных решений....11
1.3 Постановка целей и задач....13
1.4 Структура входных и выходных данных....14
Выводы....15
2. КОНСТРУКТОРСКИЙ РАЗДЕЛ....16
2.1 Функциональные требования, предъявляемые к ПМ ИДЭТ....16
2.2 Требования к надёжности....16
2.3 Требования к информационной и программной совместимости....17
2.4 Программная архитектура и алгоритм работы....18
2.5 Схема данных....22
2.6 Схема алгоритма....23
2.7 Выбор языка программирования....24
2.8 Выбор среды программирования....25
2.9 Разработка пользовательского интерфейса....39
Выводы....41
3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ....43
3.1 Программная реализация....43
3.1.1 Обмен коммерческими документами....43
3.1.2 Передача данных....49
3.2 Специализированный инструментарий....52
3.2.1 Средства работы с системами управления версиями....52
3.2.2 Средства документирования....53
3.3 Тестирование и отладка....54
3.3.1 Выбор инструментов тестирования....54
3.3.2 Особенности тестирования и отладки ПМ ИДЭТ....57

3.3.3 Результаты экспериментальной проверки.....58
Выводы.....59
ЗАКЛЮЧЕНИЕ.....60
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....61
ПРИЛОЖЕНИЕ 1. Текст программы…….……65
ПРИЛОЖЕНИЕ 2. Руководство оператора……..……114

СОКРАЩЕНИЯ

БД - База данных

БУС - 1С-Битрикс: Управление сайтом

ВКР - Выпускная квалификационная работа

ОКВЭД - Общероссийский классификатор видов экономической деятельности

ОКЕИ - Общероссийский классификатор единиц измерения

ОКП - общероссийский классификатор продукции

ОС – Операционная система

ПК - Персональный компьютер

ПМ ИДЭТ - Программный модуль интеграции данных для электронной торговли

ПО - Программное обеспечение

ТЗ - Техническое задание

ЦОД - центр обработки данных

API - application programming interface

ASCII - American standard code for information interchange

CSV - Comma-Separated Values

CMS - Content management system

DOM - Document Object Model

EDI - Electronic data interchange

HDD - hard disk drive

HTTP - HyperText Transfer Protocol

IDE - Integrated development environment

JSON - JavaScript Object Notation

MVC - Model-view-controller

RAM - Random Access Memory

RSS - Rich Site Summary

SAX - Simple API for XML

SQL - Structured query language

URL - Uniform Resource Locator

W3C - World Wide Web Consortium

XML - Extensible Markup Language

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

База данных (БД) - совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных. [9]

Операционная система (ОС) - совокупность системных программ, предназначенная для обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю определенного набора услуг. [8]

Персональный компьютер (ПК) - настольная микро-ЭВМ, имеющая эксплуатационные характеристики бытового прибора и универсальные функциональные возможности. [8]

Программный модуль (ПМ) - функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом или поименованной непрерывной её части, предназначенный для использования в других программах. [21]

Программное обеспечение (ПО) - совокупность программ регулярного применения и сопровождающей их документации, предназначенная для решения задач пользователей на компьютерной технике. [5]

Интерфейс программирования приложений (Application programming interface, API) - набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах. Используется программистами при написании всевозможных приложений. [43]

Система управления содержимым (Content management system, CMS) - информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления контентом (то есть содержимым). [32]

Интегрированная среда разработки (Integrated development environment, IDE) - комплекс программных средств, используемый программистами для разработки программного обеспечения. Среда разработки включает в себя: текстовый редактор, компилятор и интерпретатор, средства автоматизации сборки, отладчик. [14]

ВВЕДЕНИЕ

Выполнение выпускной квалификационной работы проходило в компании ООО “АПИК АЙТИ”, занимающейся разработкой программного обеспечения, в частности разработкой сайтов и корпоративных систем со сложной продуманной логикой в таких сферах, как государственные структуры, промышленность, финансы и недвижимость. Разработанный программный модуль интеграции данных между сайтами для электронной торговли (далее ПМ ИДЭТ) имеет высокую практическую значимость для решения задач работы современных интернет-магазинов, одной из которых является интеграция данных между сайтами.

Сейчас для создания своего интернет-магазина не нужно обладать специализированными знаниями, привлекать программистов, дизайнеров или аналитиков. Предприниматель получает в свои руки готовую площадку для продажи товаров через Интернет, но для полноценного функционирования такого интернет-магазина необходимо наполнить его товарами и поддерживать актуальное состояние их цен, изображений, параметров хранящихся в базе данных. Следовательно, встаёт проблема поиска простого, быстрого и мало-затратного способа поддержания, актуального состояния содержимого базы данных интернет-магазина.

Выделенные проблемы затрагивают предпринимателей-дистрибьюторов, занимающихся агрегированием и перепродажей товаров от множества поставщиков. Если же рассматривать проблему со стороны оптовых продавцов, то они заинтересованы в наращивании сети продавцов-дистрибьюторов, реализующих их товары. Для этого должны быть хорошо налажены каналы передачи массивов данных с информацией о товарах от оптовых к розничным продавцам.

Для предпринимателей актуальность затронутой темы серьёзно возросла в последние года, в виду нестабильной экономической ситуации в стране. На таком динамическом фоне заметен рост интереса к инструментам, позволяющим своевременно, сравнительно просто и дёшево обновлять содержимое базы данных, получая данные напрямую от оптовых поставщиков. А для самих поставщиков данные процессы позволят нарастить сеть продавцов-дистрибьюторов, реализующих их продукцию.

Целью данной работы является повышение частоты обновления содержимого базы данных.

Задачи ВКР:

- исследование предметной области;
- сравнительный анализ существующих программных решений;
- выбор языка и среды программирования;
- разработка схемы данных ПМ ИДЭТ;
- разработка схем алгоритмов ПМ ИДЭТ;
- разработка пользовательского интерфейса;
- программная реализация ПМ ИДЭТ;
- отладка и тестирование ПМ ИДЭТ;
- разработка руководства оператора.

Программный модуль должен обеспечивать следующие возможности:

- формирование и отправка запросов к информационной системе поставщика;
- принятие входящих запросов для взаимодействия;
- генерация и манипуляция XML-документами;
- обеспечение надёжности и целостности при пересылке данных;
- обработка и интеграция полученных данных с информационной системой;
- логирование событий и рассылка уведомлений о внештатных ситуациях.

Пояснительная записка состоит из введения, трёх разделов, заключения, списка литературы и двух приложений:

- раздел 1 содержит описание предметной области и выявленной проблемной ситуации. В нём приведён обзор существующих программных решений, сформулированы цель и задачи исследования. Также в раздел включено описание концептуальной модели предметной области;
- раздел 2 является конструкторским разделом и посвящён разработке алгоритмов и реализации решения поставленной задачи. В нём присутствует анализ существующих языков, средств и технологий разработки, излагается реализация алгоритмов, описывается процесс разработки пользовательского интерфейса;
- раздел 3 – технологический. Включает описание технологий программирования, отладки и испытаний разрабатываемого программного модуля.

В Приложении 1 размещены фрагменты исходного кода программы. Приложение 2 содержит руководство оператора.

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

Устаревшая информация не только вводит в заблуждение пользователей, но и затормаживает бизнес-процессы. Поэтому очень важно позаботиться о выборе инструментов, позволяющих провести интеграцию с системой поставщика и организовать своевременное получение актуальной информации.

Данный раздел посвящён анализу предметной области и составлению модели разрабатываемого программного модуля.

1.1 Предварительные исследования

Для содержания современного интернет-магазина используются системы управления контентом, предоставляющие интерфейс, как для сотрудников, так и для клиентов. Функционал таких систем часто уже предусматривает возможности создания, изменения и выгрузки товаров, но в настоящее время требуется новое решение.

Рассмотрим и сравним популярные CMS. Для сравнения выбраны три самых популярных систем управления контентом по версии iTrack WordPress, Joomla, «1С-Битрикс: Управление сайтом», Joomla. [28]

Таблица 1 содержит все критерии и результаты сравнения.

Система управления контентом WordPress занимает лидирующую позицию в общем рейтинге популярных CMS, охватив 31,64%. В ходе исследования было опрошено порядка 4 миллионов доменов зоны RU. На 23,7% из которых удалось однозначно определить используемую систему. Такой результат достигается несколькими факторами: бесплатное распространение, широкая специализация разрабатываемых сайтов, использованием популярного языка программирования – PHP. Другими достоинствами являются: наличие административной панели и подробной документации, присутствие возможности создания мультиязычного сайта и д.р.

Второй системой была выбрана CMS Joomla. Согласно общему рейтингу iTrack она охватывает 23,21%. По функциональным возможностям данная систему управления контентом не уступает Wordpress. Она также распространяется бесплатно, присутствует подробная документация на разных языках, реализованная административная часть сайта и многое другое.

В качестве базирующей платформы для ПМ ИДЭТ была выбрана система управления контентом «1С-Битрикс: Управление сайтом». Согласно рейтингу, доля данной CMS составляет 63,3% среди коммерческих продуктов (первое место) и 8,33% в общем рейтинге (третье место). На «1С-Битрикс: Управление сайтом» работают и готовятся к выпуску 130 000 различных веб-проектов. Среди них - сайты государственных и правительственных структур, крупных промышленных предприятий, образовательных

учреждений, СМИ, разработчиков программного обеспечения, некоммерческих организаций.

Таблица 1 – Сравнение систем управления контентом

Параметры	WordPress[1]	Joomla[2]	«1С-Битрикс: Управление сайтом»[3]
Процентная доля в общем рейтинге CMS, % [28]	31,64	23,21	8,33
Процентная доля в рейтинге коммерческих CMS, % [28]	Не участвует	Не участвует	63,30
Форма распространения	Бесплатная	Бесплатная	Коммерческая
Редакции	Выпускается в единственной редакции	Выпускается в единственной редакции	Редакции: Первый сайт, Старт, Стандарт, Эксперт, Малый бизнес, Бизнес
Специализация сайтов	Персональные блоги, образовательные порталы, интернет-магазины	Персональные блоги, образовательные порталы, интернет-магазины	Персональные блоги, образовательные порталы, интернет-магазины, корпоративные порталы
Язык программирования	PHP	PHP	PHP
Русская локализация	+	+	+
Английская локализация	+	+	+
Наличие административной части, панели	+	+	+
Возможность разработки пользовательских	+	+	+

модулей			
Возможность интеграции с системой «1С: Предприятие»	Пользовательские модули	Пользовательские модули	Штатная возможность, пользовательские модули
Мобильное приложение	-	-	Для iOS и Android
Техническая поддержка	+	+	+
Открытость исходного кода	+	+	+
Наличие документации	+	+	+

Условные обозначения:

“+” - указанная возможность присутствует;

“-” - указанная возможность отсутствует.

Источники информации:

[1] https://codex.wordpress.org/ru:Main_Page/

[2] http://joomla.ru/docs/

[3] http://www.1c-bitrix.ru/products/cms/index.php

1.2 Анализ существующих программных решений

У разрабатываемого программного модуля существуют аналоги. Таблица 2 содержит их сравнение.

В программном продукте 1С-Битрикс: Управление сайтом встроены штатные процедуры взаимодействия и поддержки двунаправленного обмена данными с программным продуктом 1С:Предприятие [13]. Важно понимать, что обмен с 1С в режиме реального времени (real-time) — многофункциональная и сложная технология, для использования которой необходимо привлечение сторонних опытных, сертифицированных специалистов, чьи услуги требуют существенных денежных вложений, что может оказаться непозволительно большой статьёй расходов в малой компании.

Существуют и другие аналоги. Так, например, компания CMS1C предоставляет свой программный модуль UNIMODULE [11], со схожим функционалом. Модуль позволяет обмениваться данными в двух направлениях – как выгружать товары и каталоги из 1С в

Интернет-магазин нажатием одной-двух кнопок, так и загружать их оттуда для дальнейших корректировок.

Следующий аналог - это программный продукт 1С:Сеть [38]. 1С-Сеть осуществляет надежную защищенную передачу коммерческих документов (Electronic Data Interchange, или EDI), таких как заказы, накладные, каталоги и т.п., между торговыми партнерами в электронном виде.

1С-Сеть объединяет информационные системы абонентов, не требуя вмешательства оператора при осуществлении обмена. Функционирование обеспечивается выделенным центром обработки данных. 1С-Сеть обеспечивает преобразование данных, что позволяет облегчить подключение разнотипных информационных систем и избежать необходимости синхронизировать справочники, предотвращает возникновение ошибок, проверяя документы на предмет соответствия их формата и содержания стандартам электронного обмена данными, а также на соответствие логике исполняемого бизнес-процесса, генерирует необходимые уведомления менеджерам при наступлении определенных событий.

Инфраструктура 1С-Сети включает несколько компонентов. Ее основа - Центр Обработки Данных (ЦОД), отвечающий высоким требованиям отказоустойчивости и производительности. Программное ядро ЦОД взаимодействует с клиентской частью при помощи Web-services. ЦОД осуществляет маршрутизацию сообщений EDI в форматах Eancom и CommerceML между абонентами 1С-Сеть, реализует протокол гарантированной доставки и гарантирует последовательность доставки сообщений. Клиентская часть 1С-Сеть - это используемая вами информационная система предприятия, например одна из конфигураций системы программ 1С:Предприятие.

С 1С-Сеть можно работать, используя обычный Интернет-браузер через Web portal EDI. Для остальных конфигураций необходимо обеспечить работу платформы 1С:Предприятие [13]. отдельные требования предъявляются к используемой операционной системе [33].

Таблица 2 - Сравнение аналогов

Параметры	Штатные процедуры БУС[1]	UNIMODULE[2]	1С-Сеть[3]	JBZoo[4]	ПМ ИДЭТ

Обмен между сайтами на БУС	-	-	-	-	+
Демо-доступ	-	-	-	+	+
Возможность полной выгрузки каталога	+	+	+	+	+
Возможность модификации	+	+	-	-	+
ОС	Windows, Linux	Windows	Windows, Linux	Windows, Linux	Windows, Linux
Стоимость, руб	От 50 000	14 000	От 10 000 в год	От 198 в месяц	От 7 000

Условные обозначения:

“+” - указанная возможность присутствует;

“-” - указанная возможность отсутствует.

Источники информации:

[1] https://www.1c-bitrix.ru/products/cms/1c/

[2] http://cms1c.ru/vasha-lyubaya-cms/

[3] http://1c-edi.ru/services.html

[4] http://jbzoo.ru/features/import-export/

Следующий аналог - это штатная возможность импорта и экспорта приложения JBZOO APP [10].Данный программное решение работает на основе системы управления содержимым сайта Joomla. Есть огромное количество настроек, которые сохраняются в профиле. Например, можно удалять или деактивировать старые записи. Соответствие файла и материалов можно делать с помощью различных ключей (например, по артикулу). Для хранения информации используется файл формата CSV.

Существуют классические способы. Они довольно широко распространены и активно используются. Например, оптовый поставщик хранит всю информацию и работает с ней в специализированном ПО 1С-Предприятие. Выгрузка каталога продукции происходит средствами 1С-Предприятие в excel-документ. Формат таких excel-документов

нигде не регламентирован и зависит, в лучшем случае, от внутренних правил компании поставщика.

1.3 Постановка целей и задач

Целью данной работы является повышение частоты обновления содержимого базы данных.

Задачи ВКР:

- исследование предметной области;
- сравнительный анализ существующих программных решений;
- выбор языка и среды программирования;
- разработка схемы данных ПМ ИДЭТ;
- разработка схем алгоритмов ПМ ИДЭТ;
- разработка пользовательского интерфейса;
- программная реализация ПМ ИДЭТ;
- отладка и тестирование ПМ ИДЭТ;
- разработка руководства оператора.

Программный модуль интеграции данных между сайтами для электронной торговли должен обеспечивать следующие возможности:

- формирование и отправка запросов к информационной системе поставщика;
- принятие входящих запросов для взаимодействия;
- генерация и манипуляция XML-документами;
- обеспечение надёжности и целостности при пересылке данных;
- обработка и интеграция полученных данных с информационной системой;
- логирование событий;
- рассылка уведомлений о внештатных ситуациях.

1.4 Структура входных и выходных данных

Входными данными являются:

- информация о товарах, хранимая в базе данных сайта;
    - наименование;
    - цена;
    - валюта;
    - артикул поставщика (внешний);
    - артикул в информационной системе (внутренний);
    - медиа информация;
        - главное изображение товара;
        - дополнительные изображения товара;

- свойства;
    - наименование;
    - значение;
    - единица измерения;
    - тип;
- данные входящего запроса;
    - иденификатор;
    - ключ отправителя;
    - сведения о запрашиваемой информации;
    - параметры передачи.

Выходные данные ПМ ИДЭТ представляют собой:

- результат интеграции данных;
- отчёт;
    - инициатор процесса;
    - время начала;
    - продолжительность;
    - статус.

Выводы

В ходе работы над исследовательским разделом ВКР, была описана предметная область, выделена проблемная ситуация, сформулированы цели и задачи исследования. Также был проведён сравнительный анализ существующих программных решений, результатом которого стало заключение о необходимости разработки собственного программного модуля, который удовлетворит потребности выбранной целевой группы пользователей.

В результате проектирования ПМ ИДЭТ была разработана структура входных и выходных данных.

2. КОНСТРУКТОРСКИЙ РАЗДЕЛ

Для разработки программного модуля были выделены специальные требования, как к функциональным возможностям модуля, так и к информационной и программной совместимости. Описан процесс разработки программной архитектуры и основных алгоритмов работы.

2.1 Функциональные требования, предъявляемые к ПМ ИДЭТ

При выработке требований к разрабатываемому программному модулю необходимо опираться на три основных положения:

- простота внедрения и использования, что позволит исключить работу высокооплачиваемых программистов-интеграторов;
- дешевизна;
- лёгкость доработки под индивидуальные нужды.

ПМ ИДЭТ выполняет функции:

- формирование запросов к системе поставщика;
- принятие входящих запросов;
- генерация и манипуляция xml-документами;
- пересылка данных между сайтами;
- обработка и интеграция полученных данных с системой;
- логирование событий;
- рассылка уведомлений о внештатных ситуациях.

Во время работы ПМ ИДЭТ не должен нагружать сайт, чтобы это не сказывалось на работе других систем и главным образом на доступность информации для пользователей.

2.2 Требования к надёжности

Для обеспечения надёжности в ПМ ИДЭТ должны быть реализованы:

- обработка ошибок при сбоях в работе транспортного канала обмена сообщениями;
- использование стандартных документированных протоколов обмена данными;
- валидация передаваемых и принимаемых данных;
- логирование событий;
- рассылка уведомлений о внештатных ситуациях;
- корректное завершение после внезапных обрывов связи.

2.3 Требования к информационной и программной совместимости

Для непосредственной работы программного модуля необходимо, чтобы на выбранном хостинге сайта смог работать продукт 1С-Битрикс: Управление сайтом. Он разработан на языке программирования PHP и может работать на любой UNIX- или Windows-платформе.

Минимальные системные требования для работы БУС:

- PHP 5.3 – 5.6;
- Apache 1.3 и выше;
- MySQL 5.0 и выше.

К обязательным параметрам системы относят:

- поддержка регулярных выражений (PerlCompatible);
- поддержка hash функций;
- поддержка функций JSON;
- установленное расширение Multibyte String;
- обработка .htaccess.

Модуль не требует установки отдельного сервера, стабильно работает практически на любом «железе», минимально нагружая интернет-магазин.

Настройка прав на сервере хостинг-провайдера может быть индивидуальна, но прежде всего должны быть установлены права на чтение/запись из скрипта для пользователя, под которым запущен веб-сервер Apache. Права на доступ к файлам сайта по умолчанию имеет значение 0644. Права на доступ к папкам сайта по умолчанию имеет значение 0755.

На сервере должна быть подключена библиотека libcurl, позволяющая получить доступ к инструменту cURL в PHP скриптах.

Пользователи ПМ ИДЭТ должны иметь навыки работы с ПК и системой управления контентом 1С-Битрикс: Управление сайтом 12 и выше. Рекомендуется иметь сертификаты об успешном завершении учебных онлайн-курсов:

- контент-менеджер;
- администратор базовый.

Техническое средство должно быть совместимо с системой управления контентом 1С-Битрикс: Управление сайтом 12 и выше.

Таблица 3 содержит минимальные требования к составу и параметрам технических средств.

Таблица 3 – Минимальные технические требования

Процессор	Одноядерный процессор с частотой 1,3 ГГц
RAM (оперативная память)	512 МБ
HDD (объём свободного места на жёстком диске)	512 МБ
Периферийные устройства	Клавиатура, мышь
Прочее	Доступ к глабальной сети с использованием технологии «Ethernet»

2.4 Программная архитектура и алгоритм работы

Bitrix Framework основан на схеме MVC [2].

MVC (Model-view-controller, «Модель-представление-контроллер») — архитектура программного обеспечения, в которой модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента, так, что модификация одного из компонентов оказывает минимальное воздействие на другие компоненты.

Шаблон MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

- модель (Model). Модель предоставляет данные (обычно для Представления), а также реагирует на запросы (обычно от Контроллера), изменяя своё состояние;
- представление (View). Отвечает за отображение информации (пользовательский интерфейс);
- поведение (Controller). Интерпретирует данные, введённые пользователем, и информирует модель и представление о необходимости соответствующей реакции.

Важно отметить, что как Представление, так и Поведение, зависят от Модели. Однако Модель не зависит ни от Представления, ни от Поведения. Это одно из ключевых достоинств подобного разделения. Оно позволяет строить Модель независимо от визуального Представления, а также создавать несколько различных Представлений для одной Модели.

Применительно к БУС паттерн MVC представляет собой:

- модель - это API;
- представление - это шаблоны;
- контроллер - это компонент.

Сплошные линии - прямые связи, пунктир - косвенные связи.

Bitrix Framework по уровням архитектуры структуру можно описать так:

- Bitrix Framework;
    - модули;
    - компоненты;
    - файлы страниц;
- сайт;
    - шаблон;
    - компоненты;
    - страница;
- компонент;
    - вызов;
    - параметры;
    - шаблон;
- страница;
    - header;
    - workarea;
    - footer.

Модуль - это модель данных и API для доступа к этим данным. Статические методы классов модуля могут вызываться в компонентах, шаблонах, других модулях. Также внутри контекста Bitrix Framework могут создаваться экземпляры классов.

Несколько десятков модулей системы содержат набор функций, необходимых для реализации какой-то глобальной, большой задачи: веб-формы, работа интернет-магазина, организация социальной сети и другие. Модули также содержат инструментарий для администратора сайта для управления этими функциями.

Компонент - это контроллер и представление для использования в публичном разделе. Компонент с помощью API одного или нескольких модулей манипулирует данными. Шаблон компонента (представление) выводит данные на страницу.

Компоненты входят в состав модулей, но решают более узкую, частную задачу — например, выводят список новостей или товаров. Вносить свои изменения в код продукта рекомендуется на уровне компонентов. Программист может модифицировать их как угодно, использовать свои наработки и использовать неограниченное число шаблонов на каждый из компонентов. На одной странице сайта может располагаться несколько компонентов, кроме того, их можно включать в шаблон сайта. Таким образом, программист имеет возможность собрать сайт как конструктор, после чего доработать необходимые компоненты для получения желаемого результата как в функциональном, так и в визуальном плане.

Страница представляет из себя PHP файл, состоящий из пролога, тела страницы (основной рабочей области) и эпилога. Формирование страницы сайта производится динамически на основе используемого шаблона страницы, данных выводимых компонентами и статической информации, размещенной на странице.

Bitrix Framework имеет модульную структуру [20]. Каждый модуль отвечает за управление определенными элементами и параметрами сайта: информационным наполнением и структурой сайта, форумами, рекламой, рассылкой, распределением прав между группами пользователей, сбором статистики посещений, оценкой эффективности рекламных кампаний и т.д.

Модули системы, главным образом, работают независимо друг от друга. Однако в целом ряде случаев функционал одних модулей основан на возможностях других. Например, модуль Торговый каталог расширяет возможности модуля Информационные блоки и позволяет выполнять настройку цен товара в зависимости от различных условий, применять к товарам наценку и скидки и т.п.

Модуль Документооборот позволяет организовать последовательную коллективную работу с содержимым модулей Информационные блоки и Управление структурой.

После установки системы список используемых модулей можно просмотреть на странице Управление модулями (Настройки > Настройки продукта > Модули) в административном разделе системы

Управление уровнем прав пользователей на доступ к модулям системы осуществляется отдельно для каждого модуля на странице его настроек. На этой же странице выполняется управление общими параметрами работы модулей.

Страница настроек конкретного модуля может иметь различное число вкладок и полей, в зависимости от функционала модуля.

Файлы модуля располагаются в папке /bitrix/modules/ID модуля/. Структура папки:

- admin/ - каталог с административными скриптами модуля;
    - menu.php - файл с административным меню модуля;
- classes/ - скрипты с классами модуля;
    - general/ - классы модуля, не зависящие от используемой базы данных;
    - mysql/ - классы модуля, предназначенные для работы только с MySQL;
    - mssql/ - классы модуля, предназначенные для работы только с MS SQL;
    - oracle/ - классы модуля, предназначенные для работы только с Oracle;
- lang/ID языка/ - каталог с языковыми файлами скриптов модуля;
- lib/ - каталог с файлами (API: классы, логика) нового ядра D7 (есть не во всех модулях);

- install/ - каталог с файлами используемыми для инсталляции и деинсталляции модуля;
- admin/ - каталог со скриптами подключающими административные скрипты модуля (вызывающие скрипты);
- js/ - каталог с js-скриптами модуля. Копируются в /bitrix/js/ID_модуля/;
- db/ - каталог с SQL скриптами для инсталляции/деинсталляции базы данных;
  - mysql/ - SQL скрипты для инсталляции/деинсталляции таблиц в MySQL;
  - mssql/ - SQL скрипты для инсталляции/деинсталляции таблиц в MS SQL;
  - oracle/ - SQL скрипты для инсталляции/деинсталляции таблиц в Oracle;
- images/ - каталог с изображениями используемыми модулем; после инсталляции модуля они должны быть скопированы в каталог /bitrix/images/ID модуля/;
- templates/ - каталог с компонентами 1.0 модуля. (Каталог сохраняется только с целью совместимости версий.);
  - ID модуля/ - каталог с основными файлами компонент;
  - lang/ID языка/ID модуля/ - в данном каталоге находятся языковые файлы компонент модуля;
- components/пространство имен/имя компонента/ - каталог с компонентами 2.0 модуля;
- panel/имя_модуля/ - содержит css и картинки для стилей административной панели, если модуль в таковых нуждается.
- index.php - файл с описанием модуля;
- version.php - файл с номером версии модуля. Версия не может быть равной нулю.
- include.php - данный файл подключается в тот момент, когда речь идет о подключении модуля в коде, в нем должны находиться включения всех файлов с библиотеками функций и классов модуля;
- default_option.php - содержит массив с именем $ID модуля_default_option, в котором заданы значения по умолчанию для параметров модуля;
- options.php - данный файл подключается на странице настройки параметров модулей в административном меню Настройки;
- prolog.php - файл может подключаться во всех административных скриптах модуля. Обычно в нем определяется константа ADMIN_MODULE_NAME (идентификатор модуля), используемая в панели управления.

Непосредственно сами классы ПМ ИДЭТ можно разделить на следующие группы:

- взаимодействие с БД;
- отправка и принятие данных;
- интеграция данных;
- вспомогательные.

2.5 Схема данных

На рисунке 1 приведена схема данных ПМ ИДЭТ.

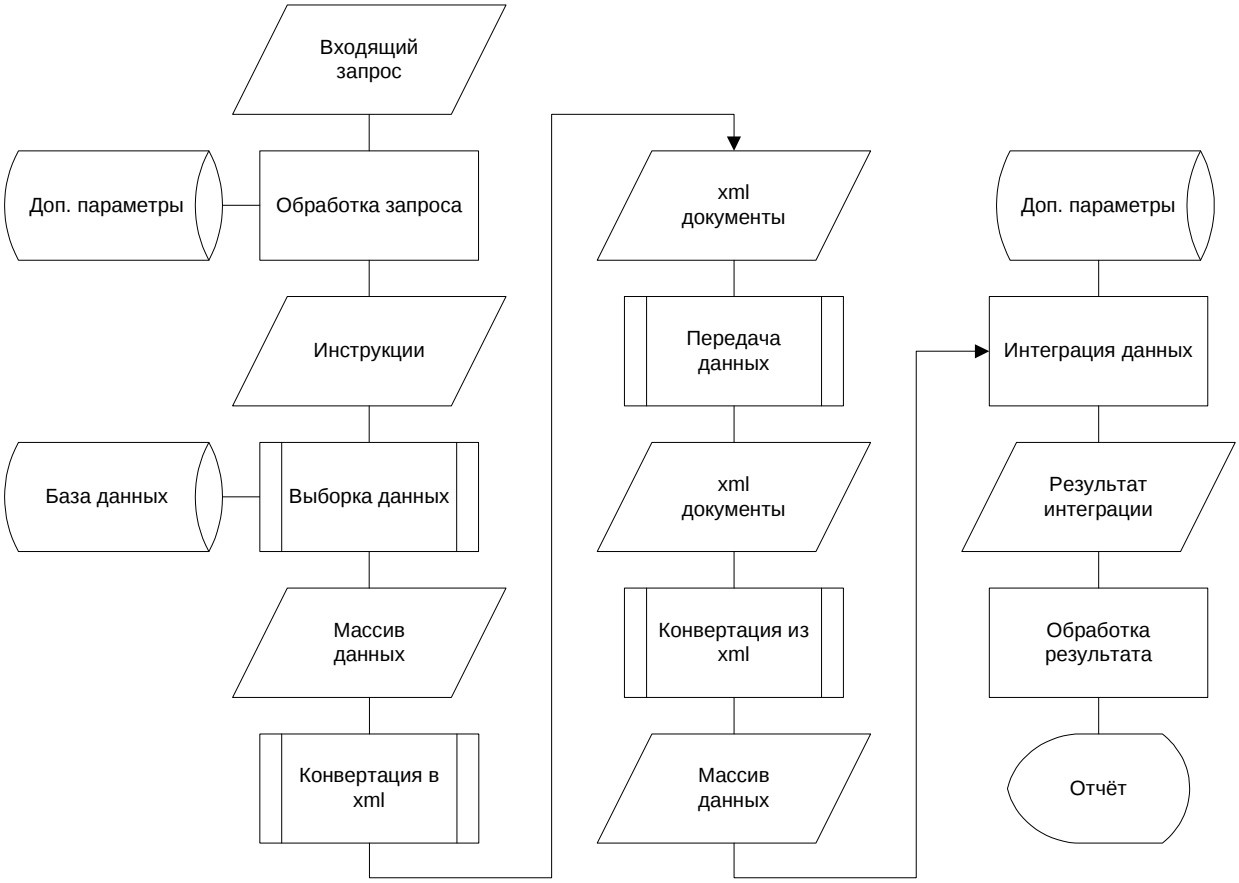

Рисунок 1 - Схема данных

Точкой входа является приходящий запрос. Во время его обработки из базы данных получаются необходимые дополнительные параметры. После этого, на основе сформированных инструкций, происходит процесс выборки данных о товарах из базы данных. До начала процесса непосредственной пересылки все данных конвертируются в документы формата XML, согласно выбранному стандарту CommerceML EDI. Данный стандарт определяет чёткие правила наименования, наличия и форматов свойств. Следующим этапом, происходит разбор пришедших данных и их интеграция в информационную систему клиента. В завершении всего процесса формируется отчёт о работе программного модуля.

В течение всего процесса происходит логирование событий.

2.6 Схема алгоритма

Схе6ма алгоритма включает такие процессы, как выборка данных, их конвертация, пересылка и непосредственно интеграция данных.

Если в ходе работы программного модуля возникли ошибки, то происходит отправка уведомлений ответственным людям.

В заключении формируется отчёт о работе программного модуля.

На рисунке 2 представлен основной алгоритм работы ПМ ИДЭТ.

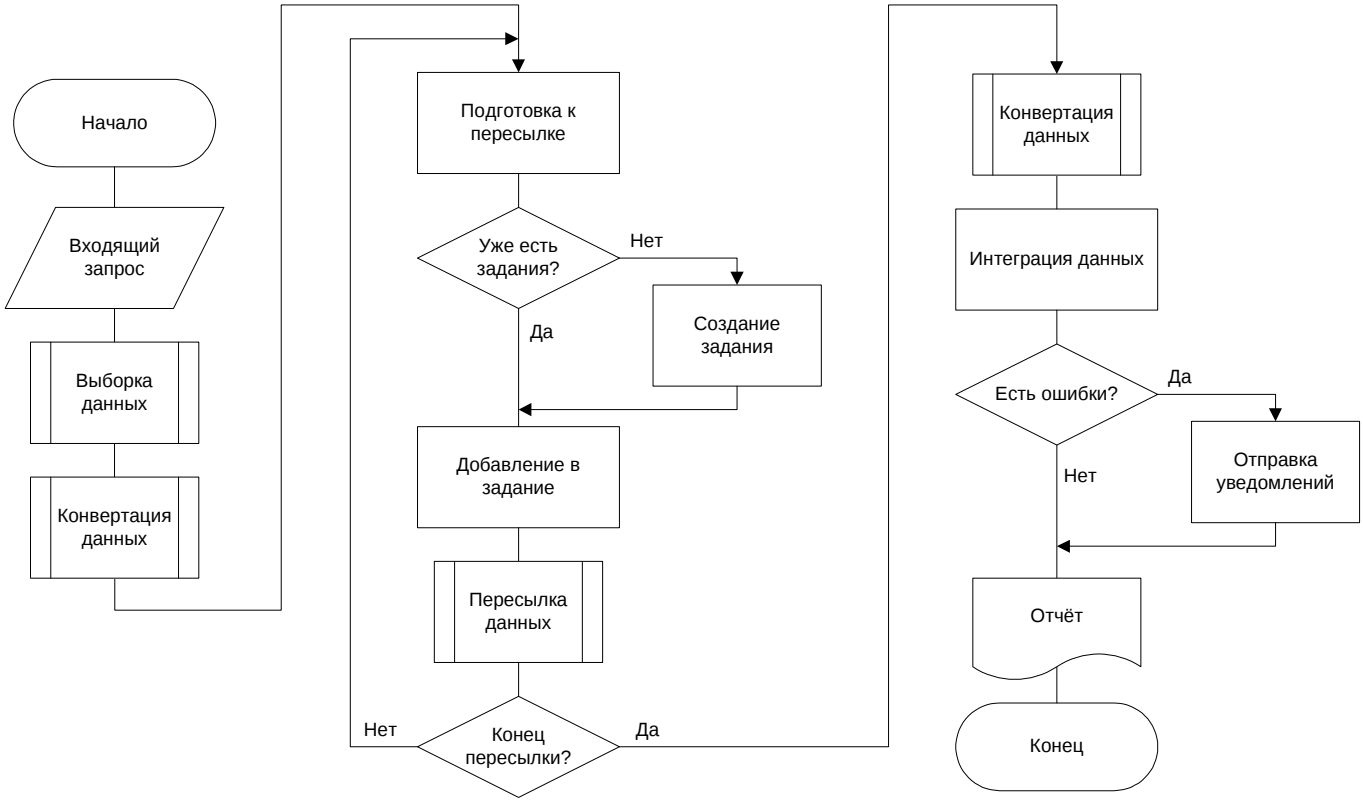

Рисунок 2 - Схема алгоритма

2.7 Выбор языка программирования

Для сравнительного анализа были выбраны следующие языки программирования: C#, C++, PHP, Java и Python. Были выделены их положительные и отрицательные стороны. Таблица 4 содержит сравнение языков программирования.

В качестве языка программирования для реализации ПМ ИДЭТ выбран PHP.

Он обладает всеми необходимыми преимуществами, как например поддержка ООП, наличие ассоциативных массивов и динамическая типизация, доступность широкого

функционала работы с различными БД и применимость для разработки под БУС. Немаловажным фактором послужило наличие опыта использования.

Таблица 4 - Сравнение языков программирования

Параметры	Язык				
Возможности языка программирования	C#[1]	C++[2]	PHP[3]	Java[4]	Python[5]
Поддержка объектно-ориентированного принципа программирования	+	+	+	+	+
Наличие ассоциативных массивов	+	-	+	-	+
Динамическая типизация	-	-	+	-	+
Наличие средств для работы с БД	+	+	+	+	+
Применим при разработке для БУС	-	-	+	-	-
Опыт использования	+	+	+	-	-

Условные обозначения:

“+” - указанная возможность присутствует;

“-” - указанная возможность отсутствует.

Источники информации:

[1] https://ru.wikipedia.org/wiki/C_Sharp/

[2] https://ru.wikipedia.org/wiki/C++/

[3] http://php.net/manual/ru/index.php

[4] https://ru.wikipedia.org/wiki/Java/

[5] http://pythonworld.ru/osnovy/vozmozhnosti-yazyka-python.html

2.8 Выбор среды программирования

В качестве среды программирования для разработки ПМ ИДЭТ были рассмотрены следующие среды: MS Visual Studio, PhpStorm, Sublime Text, NetBeans, Eclipse. Таблица 5 содержит их сравнение.

В качестве среды программирования выбран редактор Sublime Text. Решающими факторами выступили: бесплатная форма распространения, кроссплатформенность, наличие подсветки синтаксиса и опыт использования.

Таблица 5 - Сравнение сред программирования

Параметры	MS Visual Studio[1]	PhpStorm[2]	Sublime Text[3]	NetBeans[4]	Eclipse[5]
Форма распространения ПО	Коммерческая	Коммерческая	Бесплатная	Бесплатная	Бесплатная
Кроссплатформенность	-	+	+	+	+
Подсветка синтаксиса	+	+	+	+	+
Опыт работы	+	-	+	-	-

Условные обозначения:

“+” - указанная возможность присутствует;

“-” - указанная возможность отсутствует.

Источники информации:

[1] https://www.visualstudio.com/ru-ru/visual-studio-homepage-vs.aspx

[2] https://www.jetbrains.com/phpstorm/

[3] https://ru.wikipedia.org/wiki/Sublime_Text/

[4] https://ru.wikipedia.org/wiki/NetBeans/

[5] https://en.wikipedia.org/wiki/Eclipse_(software)/

В программной реализации используются возможности библиотечных средств языка PHP. Расширение socket реализует низкоуровневый интерфейс к функциям связи между сокетами [50], основанными на популярными сокетах BSD, обеспечивая возможность действовать и как сокет-сервер, и как сокет-клиент.

Более простой клиентский интерфейс доступен через функции stream_socket_client(), stream_socket_server(), fsockopen() и pfsockopen().

Сокеты как таковые [34] – мощнейший инструмент сетевого программирования. Они позволяют передавать и получать данные на прикладном уровне. Наиболее частое использование сокетов в контексте web-программирования - это работа со страницами, расположенными на разных хостах и взаимодействие с различными типами протоколов, используемых в Интернете (HTTP, FTP, SMTP, IMAP и проч.).

Для взаимодействия с сокетами в PHP чаще всего используется функция fsockopen, синтаксис которой приведён ниже.

resource fsockopen (string $host [, int $post [, int &$errno [, string &$errstr [, float $timeout]]]])

$host - хост, к которому будет производиться подключение

$port - порт, по которому будет производиться подключение

$errno - переменная, в которую запишется номер ошибки если таковая произойдёт

$errstr - переменная, в которую запишется текст ошибки если таковая произойдёт

$timeout - время в течении которого мы будем будет производиться попытка соединения с хостом

В PHP включена поддержка libcurl [44] - библиотеки функций, написанной Daniel Stenberg, которая позволяет взаимодействовать с множеством различных серверов по множеству различных протоколов. В настоящее время libcurl поддерживает протоколы http, https, ftp, gopher, telnet, dict, file и ldap. libcurl также умеет работать с сертификатами HTTPS, посылать запросы к HTTP серверам методами POST и PUT, закачивать файлы по протоколам HTTP и FTP (последнее можно сделать с помощью модуля FTP), использовать прокси-серверы, cookies и аутентификацию пользователей. Эти функции были добавлены в PHP 4.0.2.

Библиотека CURL (Client URLs) [45] позволяет передавать файлы на удаленный компьютер, используя множество Интернет протоколов. Она имеет очень гибкую настройку и позволяют выполнить практически любой удаленный запрос.

Используя CURL, web-сервер может выступать полноценным клиентом любого основанного на HTTP протоколе сервисе, к примеру: XML-RPC, SOAP, или WebDAV.

В общем виде использование библиотеки сводиться к четырем шагам [15]:

- создание ресурса CURL с помощью функции curl_init;

- естановка параметров с помощью функции curl_setopt;
- выполнение запроса с помощью функции curl_exec;
- освобождение ресурса CURL с помощью функции curl_close.

Пример использования CURL:

```
<?php
// Инициализация библиотеки curl
if ($ch = @curl_init())
{
	// Устанавливаем URL запроса
	@curl_setopt($ch, CURLOPT_URL, 'http://server.com/');
	// При значении true CURL включает в вывод заголовки
	@curl_setopt($ch, CURLOPT_HEADER, false);
	// Куда помещать результат выполнения запроса:
	@curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
	// Максимальное время ожидания в секундах
	@curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 30);
	// Установим значение поля User-agent
	@curl_setopt($ch, CURLOPT_USERAGENT, 'PHP Bot (http://blog.yousoft.ru)');
	// Выполнение запроса
	$data = @curl_exec($ch);
	// Вывести полученные данные
	echo $data;
	// Особождение ресурса
	@curl_close($ch);
}
?>
```

Пример использования GET запроса:

```
<?php
$ch = curl_init();
// GET запрос указывается в строке URL
curl_setopt($ch, CURLOPT_URL, 'http://server.com/?s=CURL');
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 30);
curl_setopt($ch, CURLOPT_USERAGENT, 'PHP Bot (http://mysite.ru)');
$data = curl_exec($ch);
curl_close($ch);
?>
```

Посылка GET запроса ничем не отличается от получения страницы. Важно заметить, что строка запроса формируется следующим образом:

http://server.com/index.php?name1=value1&name2=value2&name3=value3

где http://server.com/index.php - адрес страницы, nameX - название переменной, valueX - значение переменной.

Пример использования POST запроса

```
<?php
  $ch = curl_init();
  curl_setopt($ch, CURLOPT_URL, 'http://server.com/index.php');
  curl_setopt($ch, CURLOPT_HEADER, false);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
  // Нужно явно указать, что будет POST запрос
  curl_setopt($ch, CURLOPT_POST, true);
  // Здесь передаются значения переменных
  curl_setopt($ch, CURLOPT_POSTFIELDS, 's=CURL');
  curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 30);
  curl_setopt($ch, CURLOPT_USERAGENT, 'PHP Bot (http://mysite.ru)');
  $data = curl_exec($ch);
  curl_close($ch);
?>
```

Отправка POST запроса не многим отличается от отправки GET запроса. Все основные шаги остаются такие же. Переменные также задаются парами: name1=value1&name2=value2.

Пример HTTP-авторизации

```
<?php
  // HTTP авторизация
  $url = "http://server.com/protected/";
```

```
  $ch = curl_init();
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
  curl_setopt($ch, CURLOPT_URL, $url);
  curl_setopt($ch, CURLOPT_USERPWD, "myusername:mypassword");
  $result = curl_exec($ch);
  curl_close($ch);
  echo $result;
?>
```

Пример FTP-сессии

```
<?php
  $fp = fopen(__FILE__, "r");
  $url = "ftp://username:password@mydomain.com:21/path/to/newfile.php";
  $ch = curl_init();
  curl_setopt($ch, CURLOPT_URL, $url);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
  curl_setopt($ch, CURLOPT_UPLOAD, 1);
  curl_setopt($ch, CURLOPT_INFILE, $fp);
  curl_setopt($ch, CURLOPT_FTPASCII, 1);
  curl_setopt($ch, CURLOPT_INFILESIZE, filesize(__FILE__));
  $result = curl_exec($ch);
  curl_close($ch);
?>
```

При возникновении проблем в использовании cURL необходимо добавить следующие строки перед вызовом curl_close для получения отчета о последнем выполненном запросе:

```
<?php
  print_r(curl_getinfo($ch));
  echo "cURL error number:".curl_errno($ch)."<br/>";
  echo "cURL error:".curl_error($ch)."<br/>";
  curl_close($ch);
?>
```

Обеспечение безопасности - одно из главных требований, предъявляемых при разработке программного обеспечения. Необходимо защищать как конфиденциальные данных пользователей, так и хранимую коммерческую информацию. Часто целью злоумышленников может стать и подмена данных входящих сообщений, с целью незаконно разместить на интернет сервисах и порталах свою информацию. Для предотвращения подобных ситуаций используется комплекс мер. Рассмотрим триаду безопасности:

- аутентификация;
- целостность данных;
- конфиденциальность данных.

Цель аутентификации - доказать подлинность сторон, то есть гарантировать, что стороны бизнес-транзакции являются теми, кем они себя объявляют. Существуют несколько способов, самый простой из ни - это предоставление идентификатора пользователя и пароля. Более сложный вариант - использование сетификата X.509, который выдаётся доверенным центром сертификации. Такой сертификат включает идентификационные учётные данные и ассоциированную с ними пару из закрытого и открытого ключей. В качестве доказательства своей подлинности сторона предоставляет свой сертификат, и информацию, содержащую цифровую подпись с использованием закрытого ключа сертификата. Чтобы принимающая сторона убедилась в подлинности отправителя, подписанная информация проверяется с помощью открытого ключа, ассоциированного с сертификатом другой стороны. Процессом взаимной аутентификации называется процесс, когда происходит аутентификация обоих сторон. Данный процесс часто применяется между потребителем и поставщиком web-сервиса.

Для гарантии, что содержимое сообщения не будет изменено или повреждено при передаче через сеть Интернет, данные подписывают цифровой подписью с использованием ключей безопасности. Обеспечение целостности бизнес-информации является вторым требованием триады безопасности. Общепринятой практикой считается использование закрытого ключа сертификата X.509 отправителя для подписания цифровой подписью SOAP-тела запроса web-сервиса. Чтобы гарантировать целостность передаваемой в транзакции информации, выходящей за рамки актуального бизнес-контекста, аналогичным образом подписываются блоки SOAP-заголовка запроса. Также цифровой подписью можно подписывать и ответы web-сервисов.

Третьим требованием триады безопасности выступает обеспечение конфиденциальности. Осуществляя шифрование информации при её пересылке в запросах

и ответах web-сервисов, достигается эффект, когда данные становятся нечитаемыми для посторонних. Цель шифрования - это гарантировать, что любая попытка обратиться к данным при передаче, в памяти или после сохранения потребует соответствующих алгоритмов и ключей безопасности для дешифрования данных, без которого невозможно получить доступ к актуальной информации.

Рассмотренные концепции триады безопасности нашли своё применение при разработке ПМ ИДЭТ, используются при обмене данными между экземплярами программного модуля.

При обсуждении аутентификации, целостности и конфиденциальности используются базовые понятия: криптография, ключи, подписи и сертификаты. [3]

Криптографическая методика состоит в применении пары ключей: открытого и закрытого. Для генерации такой пары ключей используется подходящий криптографический алгоритм. Открытый ключ передаётся публичным способом любому лицу, с которым устанавливается безопасная передача сообщениями. Закрытый ключ не публикуется и держится строго в секрете. Для шифрования сообщений применяется открытый ключ, а для расшифровки соответственно закрытый. В таком случае никто кроме владельца закрытого ключа не сможет дешифровать сообщение. Данная технология получила название - асимметричное шифрование, а соответствующая пара ключей - асимметричные ключи.

Следующим методом криптографии является симметричная криптография. В данном случае для шифрования и дешифрования используется один и тот же ключ, он известен обоим участникам обмена. Производительность данного метода криптографии выше, по сравнению с асимметричным. Частым асимметричная криптография применяется для передачи общей конфиденциальной информации, а дальнейший процесс обмена данными предусматривает использование симметричной криптографии.

При защищенной передаче сообщений в сети Интернет оперируют ещё одним понятием - дайджесты сообщений. Алгоритмы дайджестов схожи с хэш-функциями: они читают данные для вычисления значения хеш-функции, называемого дайджестом сообщения. Он зависит от исходных данных и алгоритма дайджеста. Значение дайджеста использоваться для проверки целостности сообщения. Таким образом обеспечивается неизменность данных во время их передачи от отправителя к получателю. Когда получатель получает дайджест сообщения и само сообщение, то он заново производит

вычисление дайджеста. Если значения дайджеста окажутся различными при использовании одинаковых алгоритмах вычисления, то это прямое свидетельство того, что сообщение было изменено.

Дайджесты сообщений рекомендуется использовать в паре с цифровой подписью. Это решает проблему, когда и сообщение и его дайджест изменены одновременно. Идентифицировать такие случаи не представляется возможным на стороне получателя.

Алгоритм дайджеста можно применять для вычисления значения дайджеста сообщения, а затем с помощью закрытого ключа по этому значению сгенерировать цифровую подпись. Получатель, повторив вычисление дайджеста, проверяет целостность значения хэш-функции. Для проверки подлинности подписи, используется открытый ключ отправителя сообщения. При положительном результате проверки значения дайджеста и подписи можно сделать следующие выводы:

- сообщение получено от владельца открытого ключа;
- в сообщение не вносились изменения.

Цифровая подпись представляет собой структура данных, содержащую информацию о владельце сертификата и его открытый ключ. При наличии цифрового сертификата, выданного соответствующим компетентным органом, предоставляется возможность любой заинтересованной стороне убедиться в его целостности.

В дистрибутиву PHP включены три инструмента для обработки кода XML и документов XML, это так называемые специализированные API-интерфейсы:

- API-интерфейс для XML (Simple API for XML, SAX);
- объектная модель документа (Document Object Model, DOM);
- SimpleXML.

Каждый из представленных инструментов имеет свои преимущества и недостатки, которые описаны ниже.

В первую очередь можно выделить их применимость:

- для синтаксического анализа и модификации документов XML можно использовать любой из трех API-интерфейсов;
- для создания или дополнения документа XML исключительно с помощью интерфейса PHP необходимо использовать Document Object Model.

API-интерфейс Simple API for XML рассматривает документ XML как поток строковых данных. Синтаксический анализатор SAX обрабатывает документ один раз, от начала до конца, поэтому не позволяет возвращаться и выполнять действия с учетом

входных данных, находящихся в документе вслед за обрабатываемым элементом. Этот инструмент очень хорошо подходит для выполнения единообразных задач, в которых требуется применение одной и той же операции ко всем элементам определенного типа.

Расширение DOM, предусмотренное в языке PHP, позволяет считывать файл XML и создавать в памяти дерево объектов, допускающее обход. Это дает возможность начать обработку с самого документа или с любого элемента, после чего получать или задавать текстовое информационное наполнение в каждой части дерева, а также значения дочерних и родительских узлов.

Объекты Document Object Model можно либо выводить в виде текста, либо сохранять в контейнерах. В расширении DOM предусматривается формирование дерева в памяти, поэтому для обработки больших документов могут потребоваться значительные ресурсы. Наилучших результатов можно достичь, только если доступен весь документ XML. А если код XML поступает очень медленно в виде потока или требуется обработать много разных фрагментов XML как разделы одного и того же документа, то целесообразно использовать API-интерфейс SAX.

API-интерфейс SimpleXML предоставляет простой способ быстрого получения доступа к данным XML, позволяет быстро открыть файл, преобразовать некоторые из обнаруженных в нем элементов в собственные типы PHP, а затем применить к этим собственным типам необходимые операции, как и в обычной программе. API-интерфейс SimpleXML позволяет обойтись без сложностей, связанных с выполнением большого количества дополнительных вызовов, которые требуются в API-интерфейсах SAX и DOM, и ограничиться меньшим объемом памяти. Тем не менее существуют некоторые ограничения. Этот API-интерфейс некорректно работает при обработке атрибутов и глубоко вложенных элементов.

API-интерфейс для XML применяется для синтаксического анализа документов XML. По сравнению с API-интерфейсом DOM синтаксический анализатор SAX не разрабатывался под эгидой официальных организации по стандартизации. Он основан на использовании событий, вызывая указанные функции после обнаружения того, как произошло определенное событие в потоке событий. Функции-обработчики прерываний задаются с помощью кода PHP. По мере прохода анализатором по документу XML, распознаются фрагменты кода XML: элементы, внешние сущности и символьные данные. Каждый случай распознавания такого фрагмента активизируется, как событие. Если

событию задана функция-обработчик, то SAX вызывает её. После завершения работы функции обработки события синтаксический анализатор продолжает последовательную обработку документа, вызывая связанные с событиями функции до тех пор, пока не будет достигнут конец документа XML.

Так как синтаксический анализатор не может возвращаться назад и повторно обрабатывать уже проанализированные фрагменты XML, то этот процесс остается однонаправленным на протяжении всей обработки, от начала до конца документа.

Таблица 6 содержит функции API-интерфейса SAX языка XML.

Для использования API-интерфейса SAX рекомендуется выполнить последовательно следующие шаги:

1. определить события, подлежащие обработке;
2. написать функции-обработчики, для определённых на предыдущем этапе событий;
3. создать синтаксический анализатор функцией xml_parser_create()
4. вызвать функцию xml_parse() для запуска созданного синтаксического анализатора;
5. освободить память, занимаемую SAX, с помощью функции xml_parser_free().

Таблица 6 - Функции API-интерфейса SAX языка XML

Функция	Назначение
xml_parser_create ([encoding])	Создает новый экземпляр синтаксического анализатора XML и возвращает его в случае успешного завершения работы, в противном случае возвращает булево значение false. Функция принимает один необязательный параметр - идентификатор кодировки символов. Значение по умолчанию - ISO-8859-1.
xml_parse (parser, data, [final])	Вызывает на выполнение синтаксический анализатор. В качестве входных параметров используются синтаксический анализатор, документ XML в виде строки, и необязательный флаг завершения.

xml_get_error_code (parser)	Функция возвращает код ошибки, если в процессе работы синтаксического анализатора возникает проблема.
xml_error_string (errorcode)	Возвращает описание ошибки по её коду.
xml_set_element_handler (parser, start_element_handler, end_element_handler)	Устанавливает обработчики. Первый - обработчик начального дескриптора элемента, имеющий доступ к имени элемента и к ассоциативному массиву атрибутов элемента. Второй — обработчик конечного дескриптора элемента, обеспечивающего полный синтаксический анализ элемента.
xml_set_character_data_handler(parser, cd_handler)	Устанавливает функцию-обработчик, вызываемую каждый раз при обнаружении символьных данных.
xml_parser_free (parser)	Освобождает память, связанную с синтаксическим анализатором.
xml_set_default_handler (parser, handler)	Устанавливает обработчик, применяемый по умолчанию. Принимает указатель на объект синтаксического анализатора и функцию-обработчик.

API-интерфейс DOM это развитый инструмент создания, редактирования и синтаксического анализа документов XML, разработанный с учётом рекомендаций консорциума W3C.

Согласно идеологии этого интерфейса каждый документ XML рассматривается, как иерархию узлов, напоминающую ветви дерева. Исходная информация для создания дерева считывается в память из документа XML, после чего проводятся манипуляции с помощью языка PHP, а содержимое дерева записывается в другой документ XML или сохраняется в контейнере.

Основой расширения DOM служит синтаксический анализатор gnome-libxml2, благодаря меньшему потреблению ресурсов памяти.

Для использования DOM XML следует выполнить определённую последовательность шагов:

1. открыть документ XML или прочитать из оперативной памяти;

2. выполнить манипуляции;

3. вывести результирующий документ XML в виде строки или записать его в специальный файл, что позволит освободить память, используемую синтаксическим анализатором.

Таблица 7 содержит наиболее распространенные функции DOM, вызываемые в начале выполнения сценария анализа документа XML.

Таблица 7 - Функции DOM XML верхнего уровня

Функция	Назначение
domxml_open_mem()	Выполняет синтаксический анализ документа и создает объект Document. В качестве параметра принимает строку, содержащую документ XML.
domxml_open_file()	Выполняет синтаксический анализ документа и создает объект Document. В качестве параметра принимает строку, содержащую имя документа XML.
domxml_xmltree()	Создает дерево объектов PHP и возвращает объект DOM, доступный только для чтения. В качестве параметра принимает строку, содержащую документ XML.
domxml_new_doc()	Создает новый, пустой документ XML в памяти и возвращает его.

Таблица 8 содержит основные классы API-интерфейса DOM.

Таблица 8 - Классы DOM XML

Класс	Назначение
DomDocument	Содержит корневой элемент и определение Document Type Definition. Инкапсулирует документ XML.
DomNode	Содержит описание узла, который может быть корневым или любым элементом в пределах корневого элемента. Он содержит другие узлы, символьные данные и атрибуты. Инкапсулирует узел, или элемент.
DomAttr	Содержит определяемую пользователем характеристику узла (атрибут). Инкапсулирует атрибут узла.

Таблица 9 содержит основные методы классов DomDocument, DomNode и DomAttr.

Таблица 9 - Методы классов DomDocument, DomNode и DomAttr

Метод	Назначение
DomDocument -> createElement()	Создает новый элемент, дескриптором которого является переданная строка.
DomDocument -> createTextNode()	Создает новый текстовый узел.
DomDocument -> save()	Выводит документ XML из памяти в указанный файл
DomDocument -> saveXML()	Выводит документ XML из памяти в строку.
DomNode -> appendChild()	Присоединяет узел к другому узлу.
DomNode -> removeChild()	Удаляет дочерний узел.
DomAttr -> name()	Возвращает имя атрибута.
DomAttr -> value()	Возвращает значение атрибута.

Начиная с пятой версии PHP в сборку начал входить API-интерфейс SimpleXML, направленный на достижение простоты эксплуатации и сокращения потребности в памяти.

В основе данного анализатора лежит следующая идея: в PHP скрипт передается весь документ XML, после чего осуществляется его синтаксический анализ, в конце все результаты сохраняются в памяти. Документ в памяти хранится в виде собственных переменных PHP, доступных для применения.

SimpleXML в конечном итоге представляет собой характерный для языка PHP компромисс между подходами на основе Simple API for XML и Document Object Model.

В начале сценария работы с SimpleXML с помощью функции simplexml_load_string() или simplexml_load_file() считывается в виде строки или файла XML код и присваивается переменной. Таблица 10 содержит описание этих функций. Только после этого происходит непосредственная манипуляция с представлением документа XML.

Таблица 10 - Общие сведения о функциях SimpleXML

Функция	Назначение
simplexml_load_file(file)	Осуществляет импорт и выполняет синтаксический анализ файла
simplexml_load_string(string)	Осуществляет импорт и выполняет синтаксический анализ строки
simplexml_import_dom(DomDocument)	Позволяет преобразовать объект DomDocument в объект SimpleXML

2.9 Разработка пользовательского интерфейса

Для пользованием программным модулем был разработан пользовательский интерфейс, размещённый в административной части БУС.

Согласно устройству системы управления сайтом, весь интерфейс ПМ ИДЭТ помещён в специальный раздел, доступный на странице администрирования модуля.

Для работы и модерирования модуля предусмотрены три страницы:

Первая, страница со списком подключённых сайтов. Данная форма отображена на рисунке 3.

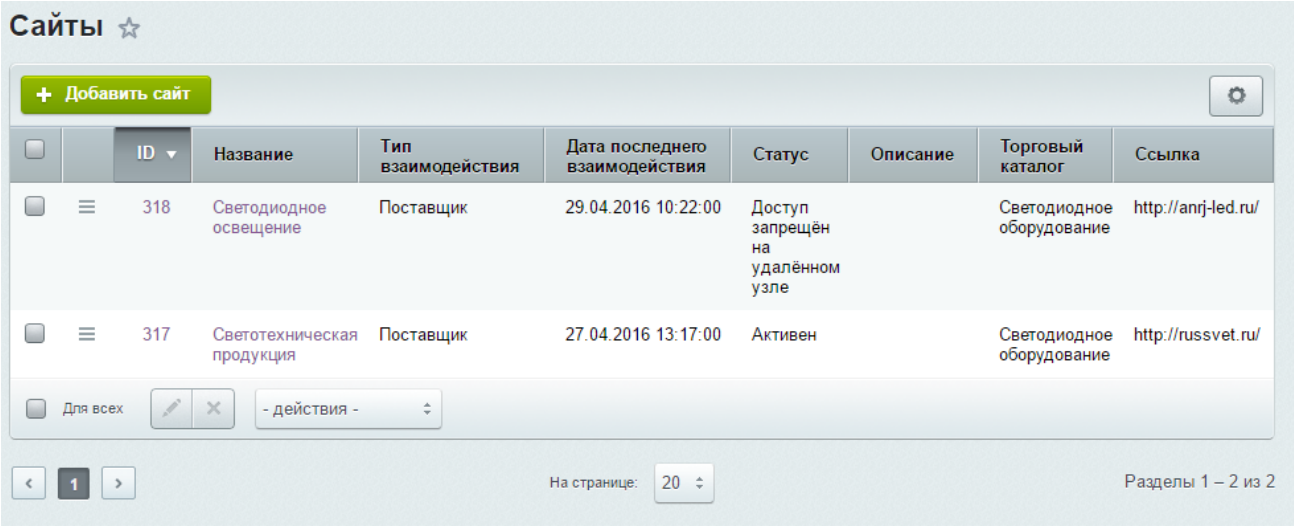

Рисунок 3 - Страница со списком подключённых сайтов

Данный интерфейс представляет собой страницу со списком всех подключённых сайтов, с которыми ведётся взаимодействие. У каждой записи есть параметры:

- идентификатор (ID);
- название;
- тип взаимодействия;
- дата последнего взаимодействия;
- статус;
- описание;
- торговый каталог;
- ссылка.

Есть возможность сортировать и фильтровать по любому параметру.

Для каждой записи можно открыть второй тип страниц - страница редактирования отдельного сайта. На ней предоставлен функционал настроек взаимодействия с удалённым узлом программного модуля. Данная форма отображена на рисунке 4.

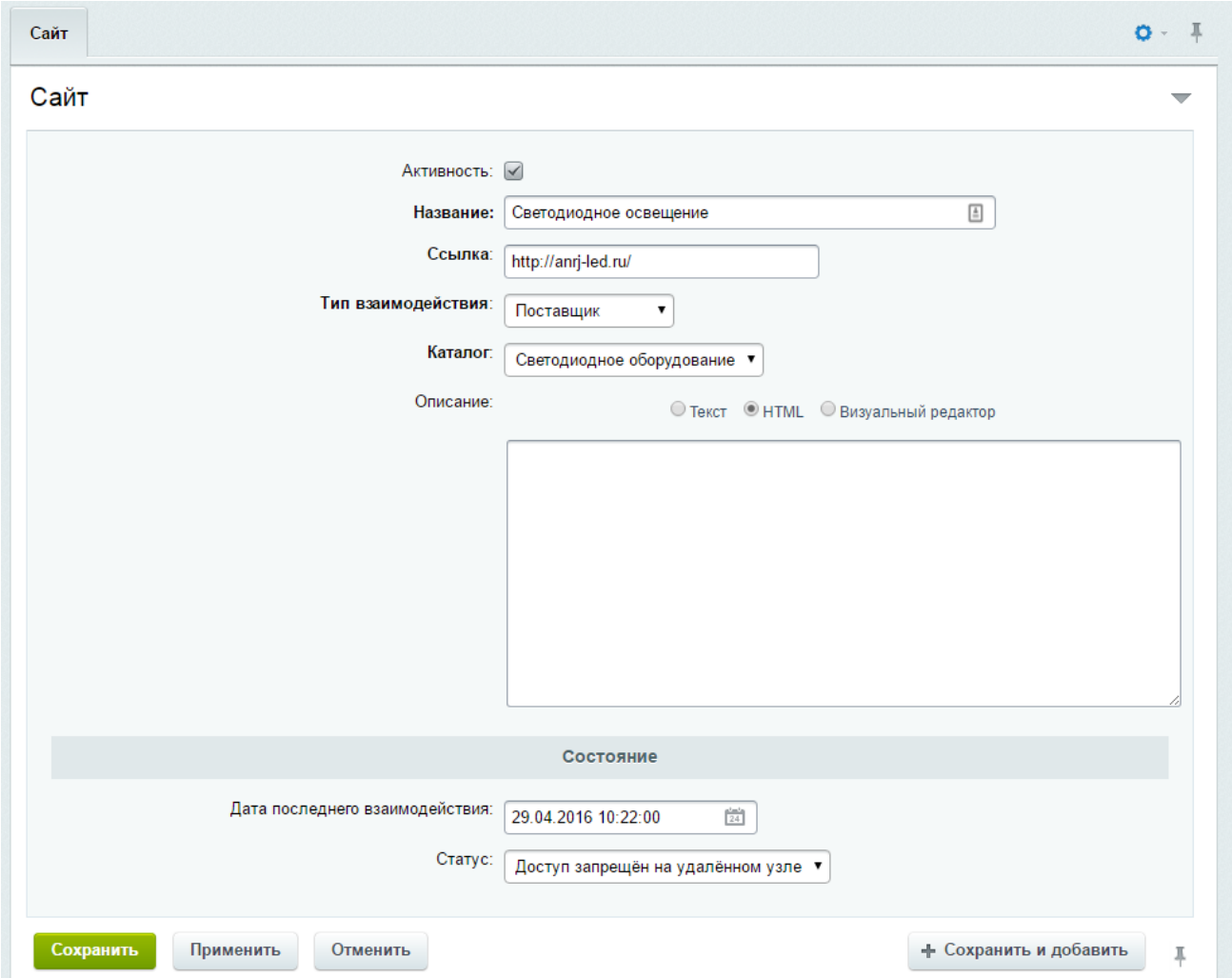

Рисунок 4 - Страница добавления или изменения существующего сайта

В данной форме можно менять поля:

- активность;
- название;
- ссылка;
- тип взаимодействия;
- каталог;
- описание.

Третьим типом страниц является страница настроек модуля. Данная форма отображена на рисунке 5.

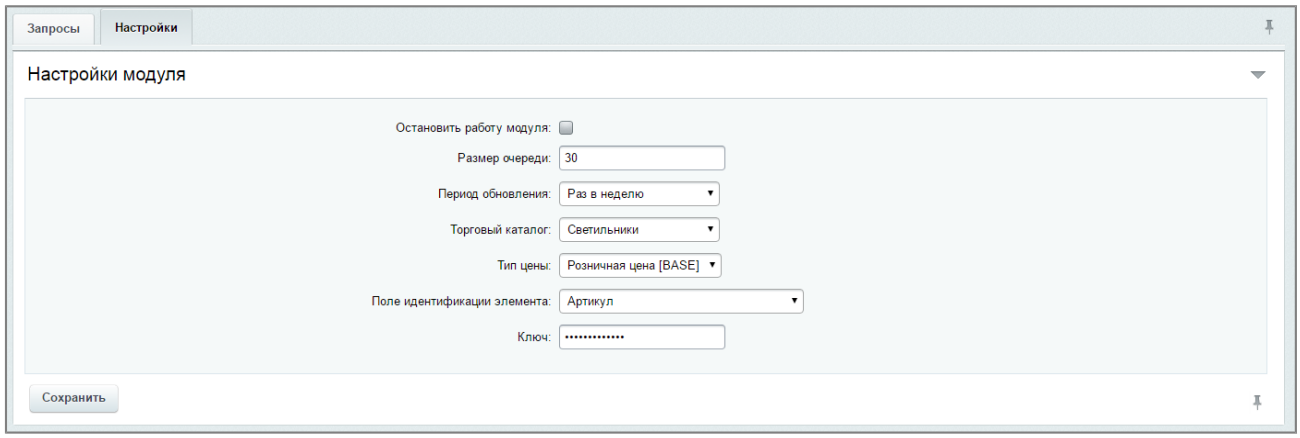

Рисунок 5 - Страница настроек собственного узла взаимодействия

На данной странице предоставлен функционал настроек работы собственного узла программного модуля и блок статистики работы.

Доступные настройки:

- остановить работу модуля;
- размер очереди;
- период обновления;
- торговый каталог;
- тип цены;
- поле идентификации элемента;
- ключ.

Выводы

В конструкторской части раздела представлены требования к надежности, информационной и программной совместимости, функциональные требования. Описаны алгоритмы работы ПМ ИДЭТ с рассмотрением общей структуры разрабатываемого программного модуля.

Был проведен сравнительный анализ основных инструментов разработки - языка и среды программирования.

В качестве языка программирования выбран серверный язык PHP. Основными его достоинствами являются поддержка ООП, наличие ассоциативных массивов, динамическая типизация и его применимость для разработки под систему 1С-Битрикс.

В качестве среды разработки выбран редактор Sublime Text. Его достоинствами являются: бесплатная форма распространения и кроссплатформенность.

3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

В технологический раздел ВРК включено описание технологий программирования, отладки, и испытаний разрабатываемого программного решения.

3.1 Программная реализация

3.1.1 Обмен коммерческими документами

Для электронного обмена коммерческими документами между информационными системами используется стандарт из линейки стандартов CommerceML [36].

Первая редакция данных стандартов была разработана при совместной работе технических специалистов фирм Extra.RU, 1С и Microsoft в 2000 году. В ходе работы было принято соглашение о поддержке и развитии единого стандарта обмена коммерческой информацией в формате XML.

За счет унификации процессов обмена коммерческой информацией, получилось добиться существенного снижения затрат на организацию информационного взаимодействия. При использовании программного обеспечения, поддерживающего данные стандарты, торговые организации с минимальными усилиями и без привлечения программистов, организовывают публикацию своих предложений на любых поддерживающих этот стандарт Web-витринах, а также обмениваются информацией между собой без специальной доработки уже внедрённых программ.

В стандарте учтены различные особенности работы как Интернет-компаний, так и торгующих организаций. Разработчики стремились обеспечить полную открытость стандартов, благодаря этому он развивался и развивается на основании объективных потребностей рынка и поддерживается широким кругом производителей экономического программного обеспечения и Интернет-компаниями. Для этого разработчики не подстраивались под особенности собственного программного обеспечения или структур информационных баз, а исходили из общих принципов организации торговой деятельности.

За основу были приняты ряд западных аналогов, однако стандарты CommerceML существенно от них отличаются, так как учитывают отечественную специфику и включают несколько универсальных решений, необходимых для российских Интернет-компаний и торговых организаций.

На данный момент линейка стандартов CommerceML состоит из трёх редакций: CommerceML 1, CommerceML 2 и CommerceML EDI.

CommerceML 1 [26] предусматривает электронный обмен следующими данными:

- каталоги товаров;
- коммерческие предложения;
- документы.

Предложение (Offer) практически совпадает с одной строкой "обычного" прайс-листа. Предложения группируются в Пакет предложений (OffersList), в котором задается общая часть всех предложений (аналог "шапки" прайс-листа).

Для того чтобы получатели предложений могли понять, какой товар предлагается, последний должен быть описан. Описание товара и его классификация "складываются" в Каталог (Catalog). Каталог может быть "внутренним", т.е. вложенным в тот же документ, что и пакет предложений, и составленным непосредственно автором пакета предложений. Он также может быть "внешним" – составленным одной из известных фирм. В этом случае в пакете предложений оговаривается, на какой каталог (классификатор) он ориентирован. Для однозначного определения товара в последнем случае достаточно ссылки (идентификатора товара во внешнем каталоге), т.е. в тот же документ, что и пакет предложений, каталог товаров можно вообще не включать. Таким образом, каталог товаров можно рассматривать как некий классификатор. Следовательно, в каталоге должен быть оговорен список Свойств (по каким критериям производится классификация). Устойчивые сочетания свойств удобно фиксировать в Наборы свойств (Profile). Для указания, какие свойства (или наборы свойств) доступны (могут быть определены, обязательно должны быть указаны) для всего каталога, для его группы или для отдельного товара, используются Ссылки на свойства (ProfileReference). Каталог (классификатор) обычно создается многоуровневым (т.е. имеющим разветвленное дерево категорий (Групп), к которым можно отнести товар). Иногда однозначная классификация может вызвать затруднения, поэтому для удобства разрешается включать товары сразу в несколько категорий. Но при этом одна из них должна быть выбрана в качестве "основной". При разработке классификаторов принято для каждой позиции указывать Аналоги (Relationship).

Указание, какими собственно свойствами из заданных в каталоге может обладать товар (или группа), достигается с помощью Ссылки на свойство (PropertyReference). При этом можно задать обязательность заполнения данного свойства. Аналогичный тип элемента создан и для набора свойств.

Для хранения значений свойств, в том числе и дополнительной, не предусмотренной классификатором информации, служит специальный тип элемента ЗначениеСвойства (PropertyValue).

Таким образом, для опубликования своего прайс-листа (составления своего пакета предложений) необходимо классифицировать свои товары.

Это можно сделать или путем составления собственного классификатора, для чего нужно:

- составить список свойств, по которым будет производится классификация;
- объединить устойчивые сочетания свойств в наборы свойств;
- составить иерархический список категорий (групп);
- отнести каждый товар к одной или нескольким категориям;
- определить для каждого товара его аналоги.

или путем нахождения своих товаров во внешнем классификаторе. Если некоторые товары не найдены во внешнем классификаторе, то для них составляется внутренний классификатор.

Следующим шагом для опубликования своего прайс-листа необходимо отправить пакет предложений.

Если при составлении пакета предложений оказалось достаточно внешнего классификатора, то отправленный файл будет содержать только пакет предложений.

Если для составления пакета (всего или его части) понадобился внутренний классификатор, то в отправляемый файл придется включить внутренний классификатор.

Вторая редакция стандарта CommerceML [42] разработана с учетом развития отрасли информационных технологий и развития языка XML. XML-схема разработана в соответствии с рекомендациями консорциума W3C, пожеланиями по расширению предыдущей редакции стандарта в части формализации описаний электронных документов и классификации передаваемых данных.

Стандарт принят и введен в действие решением совета директоров Некоммерческого партнерства "Стандарты электронного обмена информацией" 9 декабря 2003 г.

Обмен коммерческой информацией по стандарту возможен как между информационными системами контрагентов – участников торговых операций, так и между информационной системой предприятия и системой управления сайтом.

Стандарт CommerceML EDI [41] описывает документы основных бизнес-процессов взаимодействия сетевых операторов и их поставщиков, рекомендованные к применению

межрегиональной общественной организацией "Стандартизация обмена деловой информацией".

В апреле 2004 года представители ряда крупнейших российских розничных сетей и поставщиков рынка FMCG приняли решение о создании межрегиональной общественной организации "Стандартизация обмена деловой информацией". Целью работы организации является реализация эффективного информационного взаимодействия участников рынка розницы и получения ими дополнительных выгод и конкурентных преимуществ. Для этого участники договорились разработать и продвигать отраслевой стандарт информационного и технологического взаимодействия компаний.

XML-схемы стандарта CommerceML EDI включают:

- набор повторяющихся деклараций;
- файлы с XML-схемами документов конкретных бизнес-процессов.

В стандарт включено описание взаимодействия торговой розничной сети (клиента) и поставщика при предоставлении поставщиком информации о своих товарах.

Участники процесса обязаны сохранять неизменным собственный идентификатор товара на протяжении всего времени взаимодействия компаний.

Процесс обмена данными о товаре инициируется в двух случаях:

- процесс инициируется клиентом при возникновении у него необходимости получения каталога товаров поставщика. В этом случае клиент формирует документ "Запрос на каталог товаров" и отправляет его электронную версию поставщику;
- процесс инициируется поставщиком в случае, если в каталоге товаров поставщика произошли изменения, о которых необходимо сообщить клиенту.

Поставщик формирует документ, в котором содержится полный перечень предлагаемых им данному клиенту товаров и его обязательных свойств:

- идентификатор товара в кодировке поставщика;
- идентификатор товара в кодировке клиента;
- штрих-коды единицы товара;
- название товара у поставщика;
- единица измерения товара ОКЕИ;
- код товара по классификации ОКП;
- код товара по классификации ОКВЭД;
- произвольное описание товара, включающее в себя свойства, отличающие товар от другого.

Дополнительно документ может содержать необязательные для разных групп товаров атрибуты:

- название торговой марки;
- вес нетто;
- вес брутто;
- объем в $м^3$;
- количество товара в упаковке нижнего уровня;
- минимальное количество товара к поставке;
- производитель товара;
- страна происхождения товара;
- высота товара;
- количество товара в одном слое на стандартной евро-паллете (1200x800);
- срок хранения товара с момента его изготовления в днях;
- оптимальная температура хранения в градусах Цельсия;
- вид упаковки (пл. бут, жел. банка, стекло, пакет-слим и т.д.);
- вид оболочки для колбасных изд (белкозин, аметан, целлофан, синюга и т.д.);
- в случае, если набор, то перечислить весь список из набора (кондитерский или парфюмерный набор).

Электронный документ передается в информационную систему клиента, где производится его обработка, в результате которой формируется "Список принятых к работе товаров", содержащий перечень товаров из ассортимента поставщика, с которыми согласен работать клиент.

Список принятых к работе товаров в электронном виде передается в информационную систему поставщика. Список содержит коды и наименования товаров у клиента, коэффициент пересчета (отношение учетной единицы измерения поставщика к учетной единице измерения клиента).

Возможен и собственный формат документов. Например, когда придумывается и генерируется своя структура, далее вся информация сохраняется в одном из популярных файловых форматов TXT, XML или CSV. Электронный документ передается в информационную систему клиента, где производится его обработка. Самописный обмен позволяет достаточно гибко описать все его правила и алгоритмы, однако он хорошо работает при обмене небольшими объемами данных, при больших объемах возможны проблемы с производительностью.

Для работы ПМ ИДЭТ был выбран стандарт CommerceML EDI. Вследствие его характеристик, поддержки и рекомендаций компанией 1С.

Таблица 11 содержит сравнительную характеристику рассмотренных стандартов.

Таблица 11 - Сравнительная характеристика стандартов обмена данными

Характеристики	CommerceML EDI[1]	Yandex Market	Собственный

		Language[2]	формат
Формат файла	XML	YML	XML
Поддержка кириллицы	Есть	Есть	Есть
Ограничения на объём документа	Большие файлы разбиваются, размер устанавливается в параметре file_limit	500 МБ	Не накладываются
Порядок следования свойств	Не учитывается	Учитывается	Не учитывается
Доступность описания	В открытом доступе	В открытом доступе	Предоставляется по требованию
Избыточность данных	Присутствует	Присутствует	Отсутствует
Шифрование	Отсутствует	Отсутствует	Зависит от реализации

Источники информации:

[1] http://v8.1c.ru/edi/edi_stnd/90/93.htm

[2] https://yandex.ru/support/webmaster/goods-prices/technical-requirements.xml

В качестве собственного формата был выбран формат, обладающий следующими характеристиками: использование XML формата файла, поддержка кириллицы, без ограничения на объём файла, без учёта порядка следования свойств, доступность описания, отсутствие избыточности и шифрования.

3.1.2 Передача данных

Для передачи данных используются методы POST запроса, поддерживаемые HTTP протоколом. Метод запроса POST предназначен для запроса, при котором веб-сервер принимает данные, заключенные в тело сообщения, для хранения. В рамках POST запроса произвольное количество данных любого типа может быть отправлено на сервер в теле сообщения запроса. Поля заголовка в POST-запросе обычно указывают на тип содержимого. Запросы данного типа часто используются для загрузки файла или передачи

данных заполненной веб-формы. Но нельзя сказать, что каждая веб-форма должна содержать метод POST в качестве метода передачи данных по умолчанию. Многие формы используются более точно для получения информации с сервера, без изменения основных баз данных. Для таких форм поиска идеально подходит метод GET.

Метод GET предназначен для получения информации от сервера. В рамках GET-запроса некоторые данные могут быть переданы в строке запроса URI, указывающие, например, условия поиска, диапазоны дат, или другую информацию, определяющую запрос. Браузеры и веб-серверы могут иметь ограничения на длину URL, которые они обрабатывают без усечения или ошибки. URL-кодирование зарезервированных символов в адресе и строке запроса может значительно увеличить длину, в то время как HTTP-сервер Apache может обрабатывать до 4000 символов в URL, Microsoft Internet Explorer ограничивает длину любого URL 2048 символами. Равным образом, HTTP GET не должен использоваться для конфиденциальной информации, такой как имена пользователей и пароли, которые должны быть представлены вместе с другими данными для завершения запроса. Даже при использовании HTTPS, предотвращающим данные от перехвата при передаче, истории браузера и журналы веб-сервера, вероятно, содержат полные URLы в виде открытого текста, которые могут быть найдены, если система будет взломана. В этих случаях используется HTTP POST.

В документе RFC 2616 [46] рекомендуется использовать менять метод POST для любого контекста, в котором запрос не идемпотентен: то есть, он вызывает изменение состояния сервера каждый раз при выполнении, такие как отправка комментария к сообщению в блоге или интернет-голосование. На практике, метод GET часто зарезервирован, не просто для идемпотентных действий, но и для нульпотентных, то есть без побочных эффектов (в отличие от «без побочных эффектов при втором и последующих запросах» как с идемпотентными операциями). По этой причине сайты поисковых систем, таких как индексаторы поисковых систем обычно используют исключительно метод GET, для предотвращения каких-либо действий при автоматизированных запросах.

Тем не менее, есть причины почему POST используется даже для идемпотентных запросов, особенно если запрос использует не-ASCII символы или очень длинный, из-за ограничений на URL — строка запроса GET-метода может быть очень длинной, особенно при использовании URL-кодирования.

Консорциум Всемирной паутины (World Wide Web Consortium, W3C)) приводит свои рекомендации по выбору POST или GET для передачи данных [53].

Метод GET рекомендуется использовать, если взаимодействие представляет собой вопрос (то есть, это безопасная операция, такая как запрос, операция чтения или поиска).

Выбрать метод POST следует в случаях, если взаимодействие изменяет состояние ресурса таким образом, что это коснётся пользователя (например, подписка на услугу), или когда пользователь будет нести ответственность за результаты взаимодействия.

Тем не менее, для принятия окончательного решения по выбору HTTP GET или HTTP POST, необходимо исходить из практических соображений и проанализировать требования к безопасности.

Согласно выбранному стандарту обмена данными, в информационную систему продавца передаётся XML документ.

Язык Extensible Markup Language (XML) [54] можно назвать и языком разметки, и форматом хранения текстовых данных. Это подмножество языка Standard Generalized Markup Language (SGML); он предоставляет текстовые средства для описания древовидных структур и их применения к информации. XML служит основой для целого ряда языков и форматов, таких как Really Simple Syndication (RSS), Mozilla XML User Interface Language (XUL), Macromedia Maximum eXperience Markup Language (MXML), Microsoft eXtensible Application Markup Language (XAML) и open source-язык Java XML UI Markup Language (XAMJ).

Хотя язык XML вместе с базирующейся на нем платформой стандартов W3C создавался как средство представления информационных ресурсов Web, он тем не менее находит значительно более широкие применения. Назовем несколько других важных направлений, где он используется [40].

Прежде всего, создано и продолжает создаваться большое количество конкретизаций языка XML для разметки документов в различных предметных областях за счет создания DTD, согласованных различными профессиональными сообществами. Известны, в частности, версии DTD для применения в химии, географии, астрономии, истории, библиографии, издательском деле и др.

В последнее время активно развиваются технологии баз данных XML, в которых XML используется в качестве языка определения данных.

Третье направление применений XML – это системы управления документами, аналогичные тем, которые основаны на стандарте SGML и уже много лет используются на практике. Преимущество использования языка XML в этой сфере состоит в том, что становится возможной интеграции указанных систем в среду Web.

Следует далее упомянуть о применениях XML в стандартах других информационных технологий, где он используется как язык-посредник для обмена информацией между различного рода системами с помощью Web. В качестве примеров можно назвать стандарт XMI (XML Metadata Interchange) консорциума OMG обменного формата метаданных для CASE, стандарт OIM (Open Information Model) консорциума Meta Data Coalition и созданный на его основе стандарт OMG CWMI (Common Warehouse Metadata Interchange), определяющие формат представления метаданных и обмена метаданными для хранилищ данных. Планируется использовать XML для кодирования сообщений, которыми обмениваются клиент и сервер в известном стандарте ISO/IEC RDA/SQL (Remote Database Access for SQL) удаленного доступа к системам SQL баз данных. В разрабатываемом консорциумом Workflow Management Coalition (WfMC) стандарте потоков работ определяются спецификации XML DTD, позволяющие осуществлять обмен сообщениями на языке XML между программными средствами потоков работ для поддержки их интероперабельности.

В связи с успешным продвижением платформы XML в практику, начались работы над новым ранее не планировавшимся компонентом SQL/XML следующей версии стандарта языка SQL - SQL:200n. По замыслу разработчиков, он будет определять возможности совместного использования SQL и XML. В частности, будут определяться представление схем и данных SQL в форме XML-документов и наоборот.

Еще одно важное направление применения стандартов платформы XML – интеграция неоднородных информационных ресурсов.

Существует описание стандарта XML Media Types, опубликованное в документе RFC 3023 [55].

3.2 Специализированный инструментарий

3.2.1 Средства работы с системами управления версиями

На всех этапах работы над программной составляющей программного модуля интеграции данных между сайтами для электронной торговли была использована система управления версиями Git.

Система управления версиями - программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. [31]

Git – это гибкая, распределенная (без единого сервера) система контроля версий, дающая массу возможностей не только разработчикам программных продуктов, но и писателям для изменения, дополнения и отслеживания изменения «рукописей» и сюжетных линий, и учителям для корректировки и развития курса лекций, и администраторам для ведения документации, и для многих других направлений, требующих управления историей изменений. [24]

Достоинства:

- надежная система сравнения ревизий и проверки корректности данных, основанные на алгоритме хеширования SHA1 (Secure Hash Algorithm 1);
- гибкая система ветвления проектов и слияния веток между собой;
- наличие локального репозитория, содержащего полную информацию обо всех изменениях, позволяет вести полноценный локальный контроль версий и заливать в главный репозиторий только полностью прошедшие проверку изменения;
- высокая производительность и скорость работы;
- удобный и интуитивно понятный набор команд;
- множество графических оболочек, позволяющих быстро и качественно вести работы с Git’ом;
- возможность делать контрольные точки, в которых данные сохраняются без дельта компрессии, а полностью. Это позволяет уменьшить скорость восстановления данных, так как за основу берется ближайшая контрольная точка, и восстановление идет от нее. Если бы контрольные точки отсутствовали, то восстановление больших проектов могло бы занимать часы;
- широкая распространенность, легкая доступность и качественная документация;
- гибкость системы позволяет удобно ее настраивать и даже создавать специализированные контроля системы или пользовательские интерфейсы на базе git;
- универсальный сетевой доступ с использованием протоколов http, ftp, ssh и др.

Недостатки:

- UNIX – ориентированность. На данный момент отсутствует зрелая реализация Git, совместимая с другими операционными системами;
- возможные (но чрезвычайно низкие) совпадения хеш - кода отличных по содержанию ревизий;

- не отслеживается изменение отдельных файлов, а только всего проекта целиком, что может быть неудобно при работе с большими проектами, содержащими множество несвязных файлов;
- при начальном (первом) создании репозитория и синхронизации его с другими разработчиками, потребуется достаточно длительное время для скачивания данных, особенно, если проект большой, так как требуется скопировать на локальный компьютер весь репозиторий.

3.2.2 Средства документирования

Все классы ПМ ИДЭТ документированы с использованием системы phpDocumentor. Это система документирования исходных текстов на PHP. Имеет встроенную поддержку генерации документации в формате HTML, LaTeX, man, RTF и XML. Также вывод может быть легко сконвертирован в CHM, PostScript, PDF.

Прежде всего в phpDocumentor 2 отсутствует возможность работы через специализированный сайт с административным интерфейсом создания документации, включающим настройку и отслеживание процесса (лично я создавал раньше документацию именно через такой сайт, развернув его локально, как часть среды разработки). И единственным вариантом запуска остается интерфейс командной строки. Прежде всего для моих коллег, а также и для всех прочих заинтересованных лиц, опишу процесс инсталляции продукта на сервер OpenServer версии 4.6 для Windows.

phpDocumentor поддерживает PHP 5.3, что с точки зрения совместимости означает следующее: phpDocumentor может создавать документацию по массиву кода, относящегося к более ранним версиям PHP, включая классический релиз 5.2, но генерация документации должна производиться под PHP 5.3. [7]

3.3 Тестирование и отладка

3.3.1 Выбор инструментов тестирования

В PHP версии 5.6 и выше встроен интерактивный отладчик phpdbg [23]. В версиях до 5.6 нет таких средств отладки, однако существует возможность использовать внешние отладчики. Например, Zend IDE имеет встроенные инструменты отладки, а также есть несколько бесплатных расширений, такие как Xdebug, Advanced PHP Debugger или DBG. [22]

Учитывая кратковременность выполнения Web-приложений и их уровневую конструкцию, отловить ошибки в PHP-коде довольно сложно. Даже исходя из предположения, когда все уровни, за исключением PHP-кода, работают безошибочно, трассировка до обнаружения ошибки в PHP-коде является трудоёмким занятием, особенно если в приложении используется большой набор классов.

PHP-выражение echo и функции var_dump(), debug_zval_dump() и print_r() являются распространёнными средствами отладки, позволяющими выявить неоднозначные места и ошибки. Но такой способ является подходом с позиции "грубой силы". Испытывать приложение во время его работы - более эффективная стратегия. Появляется возможность систематизировать параметры запроса, просмотреть стек вызовов процедур, узнать значение переменных или объектов. Существует поддержка точек останова, позволяющих временно прервать выполнение приложения и получить уведомление об изменениях значения переменной.

Проводить такое интерактивное исследование позволяет специализированное приложение, называемое отладчиком. Он запускает или подключается к процессу для его управления и исследования памяти. В случае с интерпретируемыми языками, такими как PHP, отладчик может непосредственно интерпретировать код. Возможности современного отладчика позволяют индексировать и просматривать исходный код, отображать сложные структуры данных в человекопонятном виде, отображать состояние программы, стек вызовов, выводимые данные и значения переменных.

Инструмент Xdebug, предоставляет функциональные возможности для отображения состояния программы. Он позволяет вмешиваться в процесс для предотвращения бесконечных рекурсий, способен добавлять в сообщения об ошибках информацию о трассировке стека и функций, следить за распределением памяти.

Xdebug может обеспечить вывод подробной временной шкалы как трассировки выполнения. При разрешенной трассировке Xdebug регистрирует каждый вызов функций, включая аргументы функции и возвращаемое значение. Можно отформатировать каждый log-файл для более удобного его восприятия человеком или машиной. Первый вариант вы можете просматривать сами, а для второго можно написать отдельное приложение для анализа.

Xdebug без труда компонуется из исходных кодов в UNIX®-подобных операционных системах, в том числе Mac OS X. Если вы используете PHP в Microsoft® Windows®,

можете загрузить модуль Xdebug в бинарном виде для последних версий PHP с Web-сайта Xdebug.

Zend Debugger [16] - расширение системы Zend, тестирующее выполняющееся PHP-приложение. Загрузить и использовать Zend Debugger можно бесплатно. Однако для управления Zend Debugger и просмотра диагностической информации необходимо объединить его с клиентским приложением. Клиентское приложение может быть как элементарным, выполняющимся из командной строки, так и полнофункциональной интегрированной средой разработки (IDE), обладающей функциями редактора, завершения кода, визуальных браузеров классов и т.д.

С Zend Debugger могут взаимодействовать несколько клиентских приложений, включая PHP-плагин для Eclipse и IDE Zend Studio, предлагаемую компанией Zend Technologies.

Тестирование программного обеспечения - процесс исследования, испытания программного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

В целом существует четыре признанных уровня тестов: [37]

- Unit-тестирование;
- Интеграционное тестирование;
- Системное тестирование;
- Приёмочное тестирование.

Unit-тестирование (модульное) тестирует наименьшую единицу функциональности. С точки зрения разработчика его задачей является убедиться, что тестируемая функция выполняет именно то, для чего она реализована. Таким образом, она должна быть минимально зависима или совершенно независима от другой функции или класса. Она должна быть написана таким образом, чтобы она полностью выполнялась в памяти, т.е. она не должна подключаться к БД, не должна обращаться к сети или использовать ФС и т.д. Unit-тестирование должно быть как можно более простым.

Интеграционное тестирование "соединяет" разные единицы кода и тестирует правильно ли работают их комбинации. Он надстраивается сверху над unit-тестированием и способен отловить ошибки, которые нельзя выявить с помощью unit-тестирования, т.к. интеграционное тестирование проверяет, работает ли класс А с классом Б.

Системное тестирование создано для воспроизведения работы сценариев в условиях, приближенных к боевым. Оно, в свою очередь, надстраивается сверху над интеграционным тестированием. В то время как интеграционное тестирование обеспечивает слаженную работу различных частей системы. Системное тестирование отвечает за то, что система работает так, как предполагает пользователь, прежде чем отправить её следующий уровень.

В то время как выше приведённые тесты предназначены для разработчиков на стадии разработки, приёмочное тестирование фактически выполняется пользователями ПО. Пользователей не интересуют внутренние особенности ПО, их интересует только как работает это ПО.

Юнит-тестирование — это строительный материал для всех остальных видов тестирования. Когда мы закладываем сильную базу, мы способны построить устойчивое приложение. Однако написание тестов вручную, а также запуск тестов каждый раз когда вы вносите изменения — это процесс трудоёмкий. Если бы существовал инструмент для автоматизации этого процесса, то написание тестов стало бы гораздо более приятным занятием.

В настоящее время PHPUnit наиболее популярный фреймворк для юнит-тестирования в PHP. Кроме наличия таких возможностей, как моки (подделки) объектов, он также может анализировать покрытие кода, логирование и предоставляет множество других возможностей.

3.3.2 Особенности тестирования и отладки ПМ ИДЭТ

Отладка выполнялась для следующих категорий программных ошибок:

1. синтаксическая ошибка;
2. семантическая ошибка.

Выявление первых двух типов ошибок выполнялось на уровне среды разработки и выводимых ошибках в окно браузера при запуске PHP скриптов.

На рисунке 6 представлен фрагмент окна в процессе отладки инструментом phpdbg.

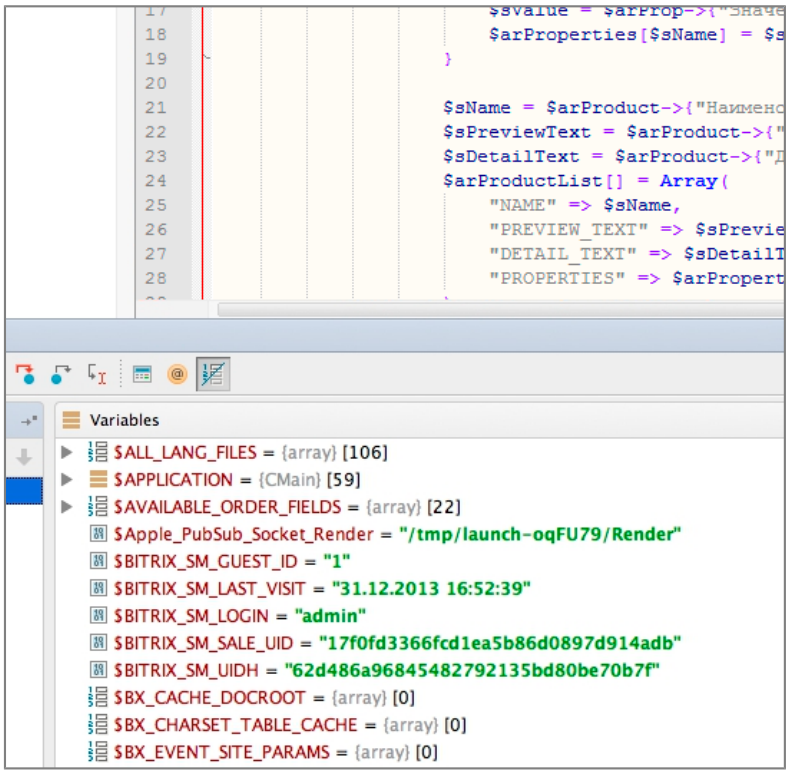

Рисунок 6 - Отладка ПМ ИДЭТ с помощью phpdbg

Тестирование ПМ ИДЭТ проводилось с использованием двух уровней:

- модульного;
- интеграционного.

Сначала выполнялись тестирование и отладка каждой функциональной части (модульное), затем проверялось функционирование всего программного модуля (интеграционное).

Модульное и интеграционное тестирование выполнялось с использованием инструмента PHPUnit.

Тестирование производительности выполнялось с помощью Xdebug. На рисунке 7 представлен фрагмент окна в процессе тестирования инструментом PHPUnit.

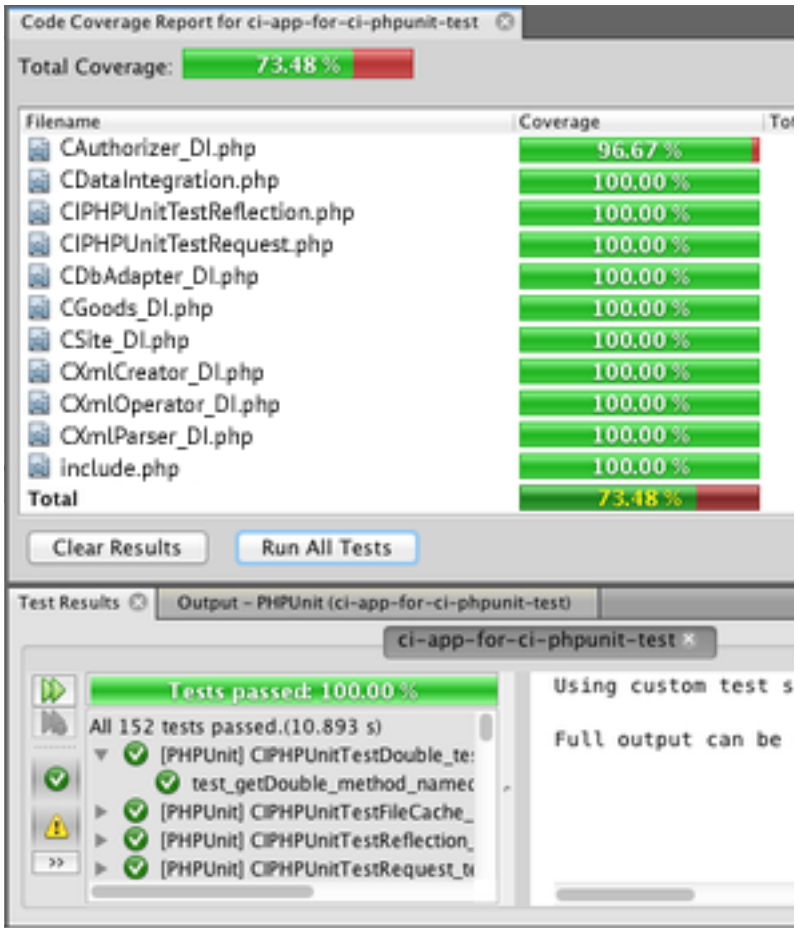

Рисунок 7 - Тестирование ПМ ИДЭТ с помощью PHPUnit

3.3.3 Результаты экспериментальной проверки

В ходе тестирования были получены следующие результаты:

1. ПМ ИДЭТ удовлетворяет функциональным требованиям, перечисленным в техническом задании;
2. Некорректной работы и аварийного завершения работы программного модуля не удалось добиться ошибками при обрыве связи;
3. Были обнаружены ошибки формирования XML документов.

Выводы

В технологическом разделе были изложены основные принципы программирования, тестирования и методы отладки, применявшиеся при разработке ПМ ИДЭТ.

Исходя из того, что программный модуль базируется на платформе БУС, он относительно просто внедряется и не требует серьёзных денежных вливаний. В результате внедрения ПМ ИДЭТ ожидается:

- уменьшение человека-часов, затрачиваемых на работу с БД;
- поддержка содержимого БД в актуальном состоянии;

- отсутствие как первоначальных, так и последующих периодических трат денежных средств.

ЗАКЛЮЧЕНИЕ

Результатом выпускной квалификационной работы стало создание рабочей версии программного модуля интеграции данных между сайтами для электронной торговли.

В конечной версии модуля реализованы все заложенные в ТЗ функции, выполнены поставленные цели и задачи.

В рамках данной разработки решены следующие задачи:

- исследование предметной области;
- сравнительный анализ существующих программных решений;
- выбор языка и среды программирования;
- разработка схемы данных ПМ ИДЭТ;
- разработка схем алгоритмов ПМ ИДЭТ;
- разработка пользовательского интерфейса;
- программная реализация ПМ ИДЭТ;
- отладка и тестирование ПМ ИДЭТ;
- разработка руководства оператора.

Программный модуль позволяет осуществлять интеграцию данных между информационными системами продавца и поставщика. Однако в отличие от существующих программных решений он позволяет осуществить связь напрямую с БУС поставщика. В результате предоставляется возможность подключиться к его системе и организовать своевременное периодическое получение актуальной информации.

Такой тип интеграции влечёт существенные выгоды:

- уменьшение человеко-часов, затрачиваемых на работу с содержимым БД;
- поддержка информации в актуальном состоянии;
- уменьшение как первоначальных, так и последующих периодических трат денежных средств на обновление содержимого БД.

ПМ ИДЭТ интегрирован в многофункциональную систему управления контентом «1С-Битрикс: Управление сайтом», предоставляющую интерфейс, как для сотрудников, так и для клиентов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Л.Г. Гагарина, Р.А. Касимов, Д.Г. Коваленко, Е.Л. Федотова, Чжо Зо Е, Б.В. Черников Методические указания по подготовке выпускной квалификационной работы по направлению подготовки бакалавров 09.03.04 «Программная инженерия»/ Под ред. док. тех. наук Б.В.Черникова. – М.: МИЭТ, 2016.

2. Архитектура продукта. [Электронный ресурс]. М., 2016. Режим доступа: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43&LESSON_ID=2817

3. Безопасность Web-сервисов. [Электронный ресурс]. М., 2016. Режим доступа: http://citforum.ru/security/internet/web_service/

4. Безопасность Web-сервисов. [Электронный ресурс]. М., 2016. Режим доступа: http://www.ibm.com/developerworks/ru/library/ws-best11/

5. Введение в программные системы и их разработку. [Электронный ресурс]. М., 2016. Режим доступа: http://www.intuit.ru/studies/courses/3632/874/lecture/14289/

6. Гагарина Л.Г., Киселёв Д.В., Федотова Е.Л. Разработка и эксплуатация автоматизированных информационных систем: учеб. Пособие / Под ред. проф. Л.Г.Гагариной. – М.: ИД «ФОРУМ»: ИНФРА-М, 2012.

7. Генератор документации phpDocumentor 2. [Электронный ресурс]. М., 2016. Режим доступа: http://appossum.com/appsite/techdetail/php-doc01/

8. ГОСТ 15971-90: Системы обработки информации. Термины и определения. – Москва: Изд-во стандартов, 1992.

9. ГОСТ Р ИСО МЭК ТО 10032-2007: Эталонная модель управления данными. – Москва: Изд-во стандартов, 2007.

10. Импорт и экспорт материалов CSV в JBZoo на Joomla. [Электронный ресурс]. М., 2016. Режим доступа: http://jbzoo.ru/features/import-export/

11. Интеграция 1С для интернет-магазина на любой CMS. [Электронный ресурс]. М., 2016. Режим доступа: http://cms1c.ru/vasha-lyubaya-cms/

12. Интеграция интернет-магазина с оптовыми поставщиками. [Электронный ресурс]. М., 2016. Режим доступа: http://my-sell.ru/integracje_z_hurtowniami/

13. Интеграция с 1С. [Электронный ресурс]. М., 2016. Режим доступа: https://www.1c-bitrix.ru/products/cms/1c/

14. Интегрированная среда разработки. [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/Интегрированная_среда_разработки/

15. Использование CURL. [Электронный ресурс]. М., 2016. Режим доступа: http://webmasterschool.ru/notes/note12.php

16. Исправление ошибок в PHP-приложениях при помощи Zend Debugger. [Электронный ресурс]. М., 2016. Режим доступа: http://www.ibm.com/developerworks/ru/library/os-php-zenddebug/
17. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / Под ред. проф. Л.Г.Гагариной. – М.: ИД «ФОРУМ»: ИНФРА-М, 2006.
18. Леонтьев В. В. Наполнение и поддержание актуальности содержимого базы данных товаров современного интернет-магазина // Молодой ученый. — 2016. — №11. — С. 189-192.
19. Микроэлектроника и информатика–2016 г. 23-я Всероссийская межвузовская научно-техническая конференция студентов и аспирантов: Материалы конференции, М.: МИЭТ, 2016.
20. Модули. [Электронный ресурс]. М., 2016. Режим доступа: https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43&CHAPTER_ID=04609&LESSON_PATH=3913.4609
21. Модульное программирование. [Электронный ресурс]. М., 2016. Режим доступа: http://ru.wikipedia.org/wiki/Программный_модуль/
22. Найдите ошибки в PHP-приложениях при помощи Xdebug. [Электронный ресурс]. М., 2016. Режим доступа: http://www.ibm.com/developerworks/ru/library/os-php-xdebug/
23. Об отладке в PHP. [Электронный ресурс]. М., 2016. Режим доступа: http://php.net/manual/ru/debugger-about.php
24. Обзор систем контроля версий. [Электронный ресурс]. М., 2016. Режим доступа: http://all-ht.ru/inf/prog/p_0_1.html
25. Обзор IDE средств для программирования. [Электронный ресурс]. М., 2016. Режим доступа: http://www.javaportal.ru/projects/taidej/results.html
26. Обмен коммерческой информацией по стандарту CommerceML, первая редакция. [Электронный ресурс]. М., 2016. Режим доступа: http://v8.1c.ru/edi/edi_stnd/90/91.htm
27. Основные сведения по Bitrix Framework. [Электронный ресурс]. М., 2016. Режим доступа: http://dev.1c-bitrix.ru/api_help/
28. Рейтинг CMS по версии iTrack. [Электронный ресурс]. М., 2016. Режим доступа: http://www.itrack.ru/research/cmsrate/
29. Роберт Басыров. 1С-Битрикс. Постройте профессиональный сайт сами!: Учебное пособие – 2009
30. Роберт Басыров. Открываем интернет-магазин с помощью 1С-Битрикс: Учебное пособие – 2009
31. Система управления версиями. [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/Система_управления_версиями/

32. Система управления содержимым. [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/Система_управления_содержимым/
33. Системные требования 1С:Предприятия 8. [Электронный ресурс]. М., 2016. Режим доступа: http://v8.1c.ru/requirements/
34. Сокеты в PHP. [Электронный ресурс]. М., 2016. Режим доступа: http://forum.vingrad.ru/articles/topic-103996.html
35. Сравнение PHP IDE. [Электронный ресурс]. М., 2016. Режим доступа: http://rmcreative.ru/playground/PHP_IDE.pdf
36. Стандарты электронного обмена коммерческой информацией CommerceML. [Электронный ресурс]. М., 2016. Режим доступа: http://v8.1c.ru/edi/edi_stnd/90/
37. Тестирование программного обеспечения. [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/Тестирование_программного_обеспечения/
38. Услуги 1CNet. [Электронный ресурс]. М., 2016. Режим доступа: http://1c-edi.ru/services.html
39. Федотова Е.Л., Портнов Е.М. Прикладные информационные технологии: учебное пособие / Е.Л. Федотова, Е.М. Портнов. – М.: ИД «ФОРУМ»: ИНФРА-М, 2013.
40. Функциональные возможности и направления использования стандартов платформы XML. [Электронный ресурс]. М., 2016. Режим доступа: http://www.elbib.ru/index.phtml?page=elbib/rus/methodology/xmlbase/xml_conf/
41. Электронный обмен данными в цепочке поставок, стандарт CommerceML EDI. [Электронный ресурс]. М., 2016. Режим доступа: http://v8.1c.ru/edi/edi_stnd/90/93.htm
42. Электронный обмен данными по стандарту CommerceML, вторая редакция. [Электронный ресурс]. М., 2016. Режим доступа: http://v8.1c.ru/edi/edi_stnd/90/92.htm
43. Application programming interface. [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/API/
44. cURL and libcurl. [Электронный ресурс]. М., 2016. Режим доступа: https://curl.haxx.se/
45. cURL. [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/CURL/
46. Hypertext Transfer Protocol. [Электронный ресурс]. М., 2016. Режим доступа: https://tools.ietf.org/html/rfc2616/
47. Jane Mitchell. PHP Web Services: Учебное пособие – 2013
48. NetBeans. [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/NetBeans/
49. PHP и XML. [Электронный ресурс]. М., 2016. Режим доступа: http://addphp.ru/materials/base/1_21.php
50. PHP: Введение - Manual. [Электронный ресурс]. М., 2016. Режим доступа: http://php.net/manual/ru/intro.sockets.php

51. POST (HTTP). [Электронный ресурс]. М., 2016. Режим доступа: https://ru.wikipedia.org/wiki/POST_(HTTP)/
52. Request for Comments [Электронный ресурс]. – RFC 2616 - Hypertext Transfer Protocol - HTTP/1.1 – URL: https://tools.ietf.org/html/rfc2616/
53. URIs, Addressability, and the use of HTTP GET and POST. [Электронный ресурс]. М., 2016. Режим доступа: https://www.w3.org/2001/tag/doc/whenToUseGet.html
54. XML для PHP-разработчиков. [Электронный ресурс]. М., 2016. Режим доступа: https://www.ibm.com/developerworks/ru/library/x-xmlphp1/
55. XML Media Types. [Электронный ресурс]. М., 2016. Режим доступа: https://tools.ietf.org/html/rfc3023/