

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Базовая кафедра автоматизации бизнес-процессов

ОТЧЕТ

по лабораторной работе

«Разработка клиент-серверного приложения для складского учета с использованием
REST интерфейса»

Выполнил:

студент группы ПИ 155-2

Попович М.М.

Проверил:

ассистент

Красиков В.Е.

Содержание

1. Используемые технологии при разработке.	3
2. Постановка задачи.....	5
2.1. Описание приложения	5
2.2. Разработка структуры базы данных	5
2.3. Создание БД в Microsoft SQL Server	6
2.4. Размещение конфигурации на локальном сервере.	7
2.5. Создание REST интерфейса OData для интеграции с платформой .NET	7
2.6. Разработка клиентского приложения	9
Заключение.....	12

1. Используемые технологии при разработке.

Структура базы данных реализовывалась с помощью инструмента «1С: Предприятие 8.3». При использовании стандартной «файловой» СУБД, возникает следующие ряд проблем:

- Отсутствие безопасности. Любой пользователь системы может скопировать файл БД.
- Малая масштабируемость системы — в некоторых случаях система начинает работать медленно уже при 5-7 пользователях.
- Некоторые функции программы не работают в файловом режиме (например, регламентные задания).
- Ограничена в размере (около 4-12 гб).

Поэтому, в качестве СУБД был выбран Microsoft SQL Server, которые обладает следующим рядом преимуществ:

- Масштабируемость. Использование СУБД MSSQL даст возможность расширять количество пользователей, создавать большое количество баз и, при выделении дополнительного количества аппаратных ресурсов, это не приведет к торможению базы и блокировкам данных.
- Отказоустойчивость. Реализация базы на СУБД дает возможность использовать технологии кластеризации, что обеспечит горячее резервирование работы с 1С.
- Обслуживание. В СУБД MS SQL есть возможность настроить автоматические регламентные операции, которые будут периодически оптимизировать работу с базой.
- Резервирование. Благодаря использованию СУБД, возможно реализовать более надежную систему резервного копирования средствами самого SQL, также резервное копирование лога СУБД дает возможность восстановить данные с точностью до транзакции.
- Мониторинг. MS SQL имеет ряд датчиков, отслеживая которые заранее возможно увидеть проблемы с производительностью, безопасностью, а также, отслеживать успешное выполнение операций оптимизации и резервного копирования.

- Использование для других приложений. MS SQL является одной из самых популярных СУБД, большинство приложений использующих СУБД работают с MS SQL. Таким образом внедрение MS SQL для 1С может служить подготовкой почвы для внедрения или оптимизации других приложений.

Платформа 1С также позволяет использовать в качестве СУБД сторонних поставщиков: PostgreSQL, IBM DB2, Oracle Database.

REST интерфейс, был также сформирован с помощью стандартных средств платформы 1С, позволяющих автоматически формировать всю структуру для прикладного решения. При формировании REST интерфейса использовался протокол OData 3.0. Это открытый веб-протокол для запроса и обновления данных. Он позволяет оперировать данными, используя в качестве запросов HTTP-команды. Получать ответы можно в форматах: Atom/XML и JSON.

Клиентское приложение – это веб-приложение ASP.NET (.NET Framework). Для удобства работы с REST интерфейсом использовалась открытая библиотека RestSharp. Он позволяет упростить взаимодействие с API-интерфейсами и обрабатывать различные HTTP-запросы.

Веб-сервер для размещения серверной части – Apache 2.2.

2. Постановка задачи

2.1. Описание приложения

Тематика разрабатываемого приложения – «Складской учет».

Требуется реализовать учет номенклатур внутри предприятия. Номенклатуры перемещаются между сотрудниками, кабинетами, складами. Предприятие принимает номенклатуры от поставщиков, а также списывает старые. Ход движения номенклатуры должен отслеживаться по сотрудникам, кабинетам, отделам и складам.

2.2. Разработка структуры базы данных

На основании поставленной задачи, с помощью методологии IDEF1x была разработана структура базы данных (см. рис. 1)

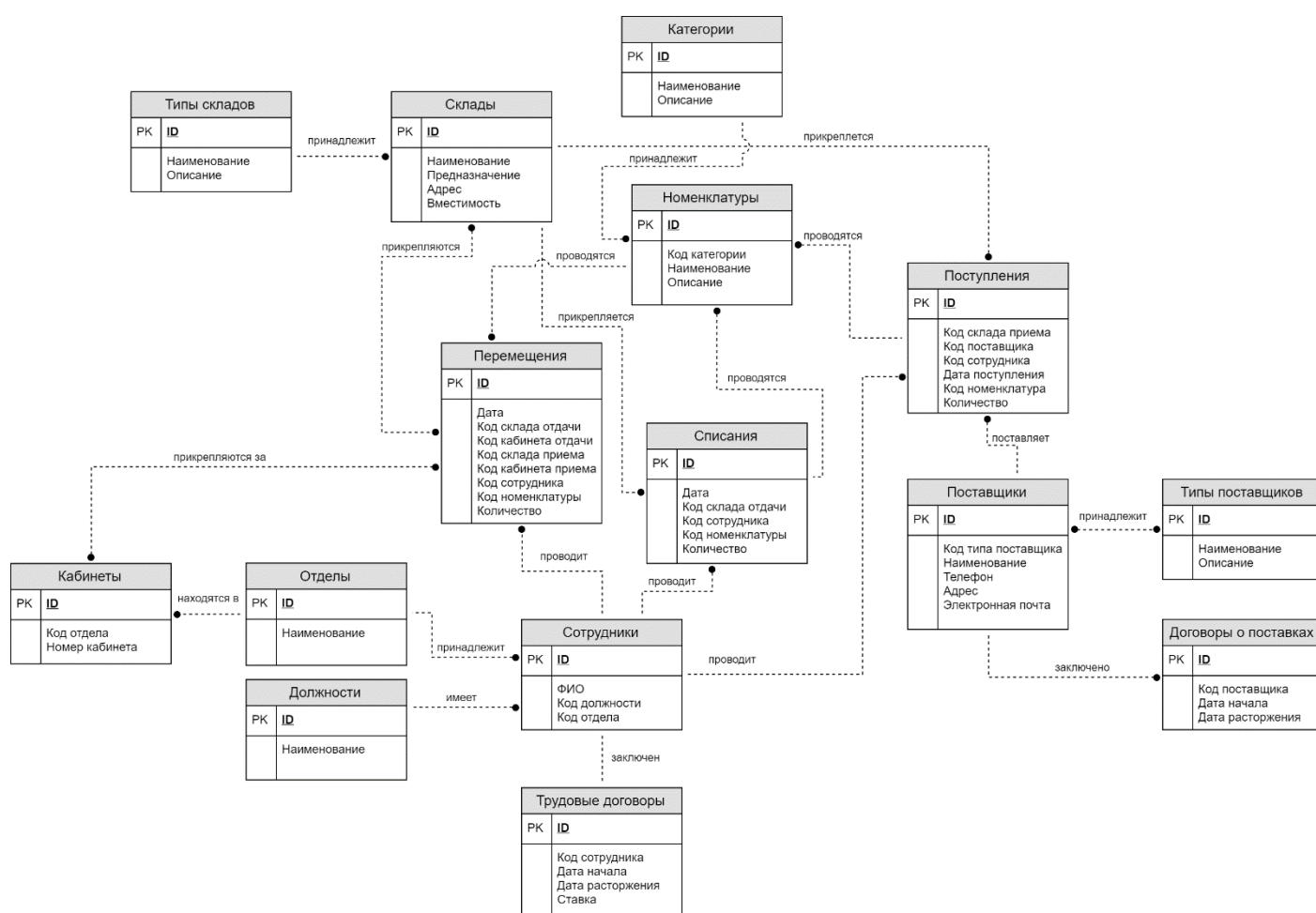


Рисунок 1 – Структура базы данных (IDEF1x)

В системе используется 15 таблиц, среди которых 10 являются справочниками и 5 документами.

2.3. Создание БД в Microsoft SQL Server

Как уже было сказано ранее, в качестве СУБД был выбран Microsoft SQL Server и создание структуры/связей было реализовано с помощью платформы 1С: Предприятие 8.3 (см. рис. 2).

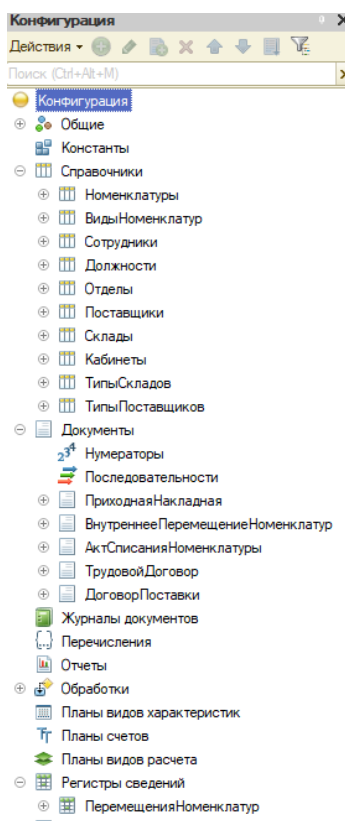


Рисунок 2 – Объекты базы данных

В СУБД MS SQL структура базы выглядит иначе. Все справочники именуются с дополнительной приставкой «Reference», документы – «Document». Для табличных частей документов используются отдельные таблицы. Пример отображения справочника «Номенклатуры» в виде таблицы MS SQL представлен на рисунке 3.

	_IDRRRef	_Version	_Marked	_PredefinedID	_Code	_Description	_Fld16	_Fld17	_Fld18RRRef
▶	оичные данные>	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000001	СКАТ Универс...	POOI4P12O41P4	СКАТ Универс...	<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000002	СКАТ Кредо 20...	235329I523523I5	СКАТ Кредо 20...	<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000003	55 Samsung UE...	OI3U4OI32U523...	HD Ready 1366...	<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000004	24" Samsung U...	UE24H4080AUX...	Ultra HD 4K 384...	<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000005	Стол компьют...	325O2I6PO2I36	Материал: ДСП	<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000006	Стол журналъ...	O12I4P124IP1244	Материал: сте...	<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000007	Шкаф-купе	23OI5U23I5U23...	Материал: пла...	<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000009	ASUS X556UQ-...	346YUI3Y42O3...		<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000010	ASUS X540NA-...	43IOU634OIP6U...		<Двоичные да...
	<Двоичные да...	<Двоичные да...	<Двоичные да...	<Двоичные да...	000000011	ASUS X507MA-...	32T23T23T23T2...		<Двоичные да...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3 – Пример отображения справочника «Номенклатура» в MS SQL

2.4. Размещение конфигурации на локальном сервере.

Так как в качестве СУБД был выбран MS SQL Server, то потребовалось развернуть локальный сервер для хранения используемой конфигурации. Настройки локального сервера представлены на рисунке 4.

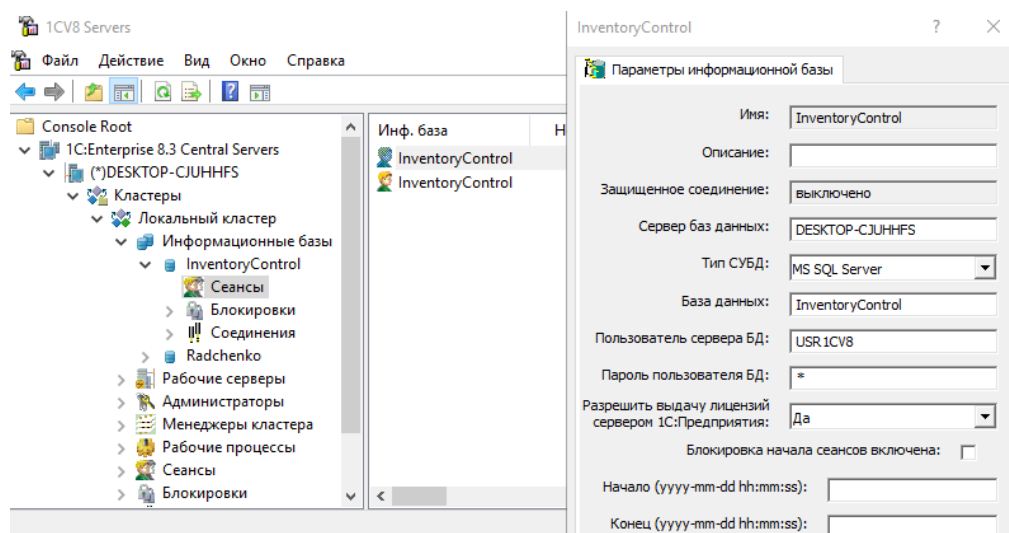


Рисунок 4 – Настройки размещения конфигурации на локальном сервере

2.5. Создание REST интерфейса OData для интеграции с платформой .NET

Для публикации созданной базы данных на веб-сервере использовались также стандартные возможности платформы «1С: Предприятие 8.3». В качестве веб-сервера для размещения серверной части использовался Apache 2.2. На рисунке 5 представлены настройки публикации REST-сервиса.

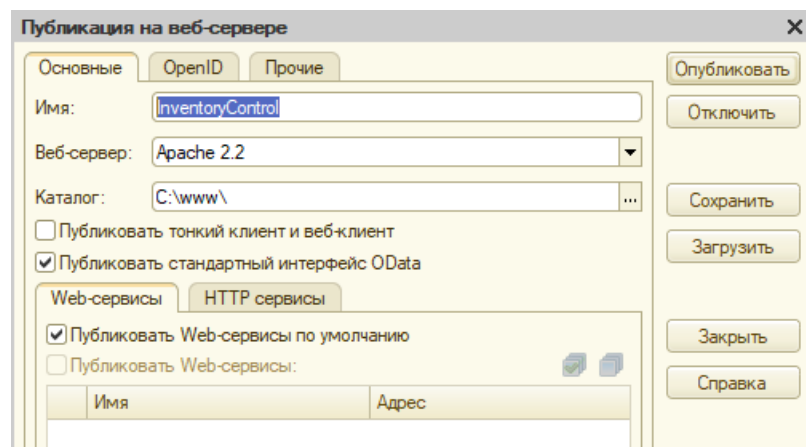


Рисунок 5 – Настройки публикации REST-сервиса на веб-сервере

В сформированном REST интерфейсе доступны реквизиты объектов конфигурации, доступны операции создания, чтения, модификации и удаления данных, а также некоторые методы встроенного языка. Например, для документов: Post() – проведение документа, для регистров сведений: SliceLast() и SliceFirst() – срез последних и срез первых.

Чтение осуществляется с помощью GET запросов. Например:

GET /InventoryControl/odata/standard.odata/Имя_ресурса

Создание нового элемента данных выполняется POST-запросом. В качестве значения ссылки передается нулевой GUID. При создании и модификации объектов значения свойств передаются в теле запроса в формате XML:

POST /InventoryControl/odata/standard.odata/Имя_ресурса

Модификация существующих данных выполняется PATCH-запросом.

PATCH /InventoryControl/odata/standard.odata/Имя_ресурса(guid'%guid%')

Для удаления данных используется DELETE-запрос:

DELETE InventoryControl/odata/standard.odata/Имя_ресурса(guid'%guid%')

Вывод возможен в JSON и Atom. Возможно использование фильтров в запросе, стандартных SQL и Linq операций. Также доступны функции для работы с датами, строками и числами.

Например, получение списка номенклатур выглядит следующим образом (см. рис. 6).


```

{
  "odata.metadata": "http://localhost/InventoryControl/odata/standard.odata/$metadata#Catalog_Номенклатуры",
  "value": [
    {
      "Ref_Key": "d97e6e15-d6b0-11e8-be9c-7429af6675c4",
      "DataVersion": "AAAAAAAAAF+w=",
      "DeletionMark": false,
      "Code": "00000001",
      "Description": "СКАТ Универсал 1",
      "СерийныйНомер": "P00I4P12041P4",
      "Описание": "СКАТ Универсал 2017 - (1), как становится понятно из названия, этот компьютер является универсальным решением для выполнения различного с\у\зкоспе",
      "ВидНоменклатуры_Key": "6c2d3e17-cbe4-11e8-be9a-7429af6675c4",
      "Predefined": false,
      "PredefinedDataName": "",
      "ВидНоменклатуры@navigationLinkUrl": "Catalog_Номенклатуры(guid'd97e6e15-d6b0-11e8-be9c-7429af6675c4')/ВидНоменклатуры"
    },
    {
      "Ref_Key": "f91004ad-d6b0-11e8-be9c-7429af6675c4",
      "DataVersion": "AAAAAAAAAF+o=",
      "DeletionMark": false,
      "Code": "00000002",
      "Description": "СКАТ Кредо 2018",
      "СерийныйНомер": "235329I523523I5",
      "Описание": "СКАТ Кредо 2017 - (5) - игровой компьютер среднего ценового сегмента, который отлично подойдет для комфортной игры в современные компьютерные мате\rинсы",
      "ВидНоменклатуры_Key": "6c2d3e17-cbe4-11e8-be9a-7429af6675c4",
      "Predefined": false,
      "PredefinedDataName": "",
      "ВидНоменклатуры@navigationLinkUrl": "Catalog_Номенклатуры(guid'f91004ad-d6b0-11e8-be9c-7429af6675c4')/ВидНоменклатуры"
    },
    {
      "Ref_Key": "0f6036f3-d6b1-11e8-be9c-7429af6675c4",
      "DataVersion": "AAAAAAAAAF+Q=",
      "DeletionMark": false,
      "Code": "00000003",
      "Description": "55\\ Samsung UE55NU7100",
      "СерийныйНомер": "OI3U40I32U52325",
      "Описание": "HD Ready 1366x768",
      "ВидНоменклатуры_Key": "ba068d40-d6ad-11e8-be9c-7429af6675c4",
      "Predefined": false,
      "PredefinedDataName": "",
      "ВидНоменклатуры@navigationLinkUrl": "Catalog_Номенклатуры(guid'0f6036f3-d6b1-11e8-be9c-7429af6675c4')/ВидНоменклатуры"
    },
    {
      "Ref_Key": "1ef5ab29-d6b1-11e8-be9c-7429af6675c4",
      "DataVersion": "AAAAAAAAAF+M=",
      "DeletionMark": false,
      "Code": "00000004",
      "Description": "24\\ Samsung UE24N4080 ",
      "СерийныйНомер": "UE24N4080AUXRU",
      "Описание": "Ultra HD 4K 3840x2160",
      "ВидНоменклатуры_Key": "ba068d40-d6ad-11e8-be9c-7429af6675c4",
      "Predefined": false,
      "PredefinedDataName": "",
      "ВидНоменклатуры@navigationLinkUrl": "Catalog_Номенклатуры(guid'1ef5ab29-d6b1-11e8-be9c-7429af6675c4')/ВидНоменклатуры"
    }
  ]
}

```

Рисунок 6 – Вывод списка номенклатур с помощью OData

2.6. Разработка клиентского приложения

Клиентского приложение разрабатывалось на ASP.NET (.NET Framework). В качестве стандартного стиля был использован открытый шаблон Bootstrap. Главная страница приложения изображена на рисунке 7.

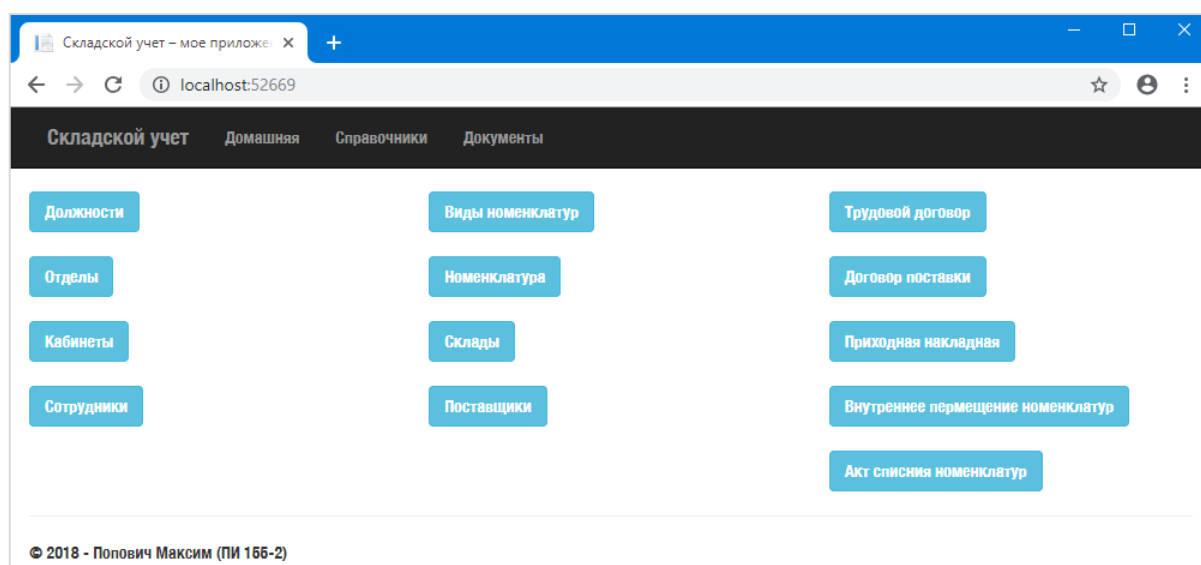


Рисунок 7 – Главная страница веб-приложения

Работа с каждым справочником/документом осуществляется на отдельной странице. Например, работа со справочником «Сотрудники» представлена на рисунке 8

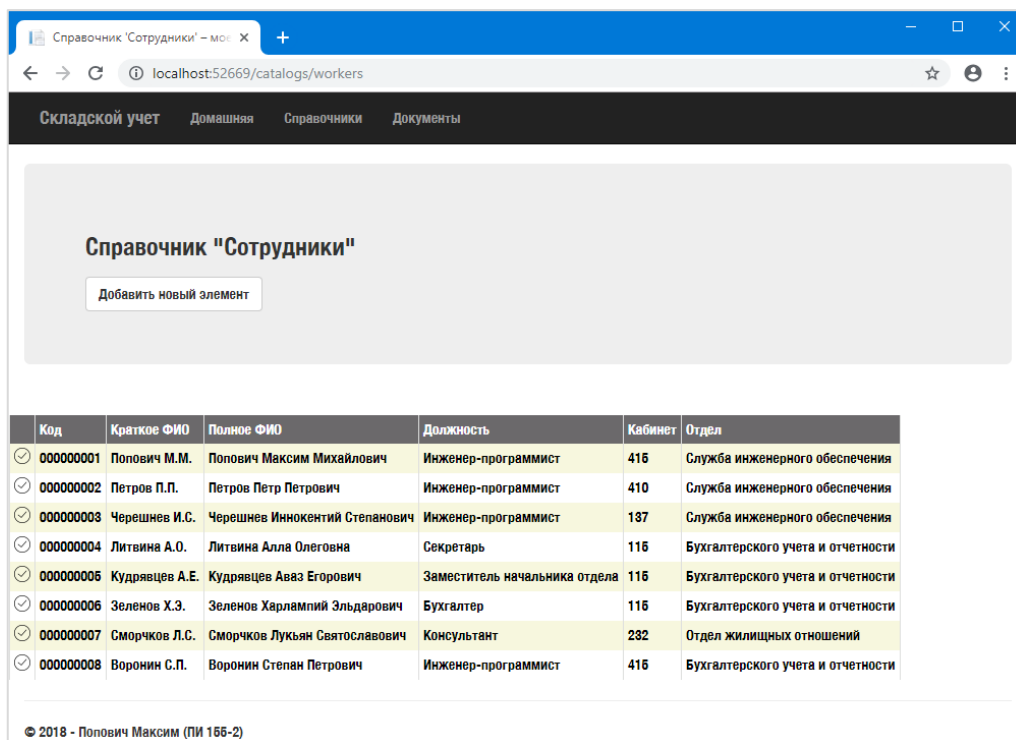


Рисунок 8 – Страница для работы со справочником «Сотрудники»

Пользователь может создать новый справочник (см. рис. 9). С помощью выпадающих списков, пользователь выбирает ссылочные типы данных.

Формуляры для добавления нового сотрудника:

ФИО:

Должность:

Кабинет:

Отдел:

Добавить Отмена

Код	Краткое ФИО	Полное ФИО	Должность
✓ 000000001	Попович М.М.	Попович Максим Михайлович	Инженер-программист

Рисунок 9 – Добавление нового сотрудника

Модификация или удаление сотрудника осуществляется путем выбора сотрудника в таблице (см. рис. 10).

ФИО
 Петров Петр Петрович
 Должность
 Инженер-программист
 Кабинет
 410
 Отдел
 Служба инженерного обеспечения

Редактировать Удалить Отмена

	Код	Краткое ФИО	Полное ФИО	Д
✓	000000001	Попович М.М.	Попович Максим Михайлович	Ин
✓	000000002	Петров П.П.	Петров Петр Петрович	Ин

Рисунок 10 – Модификация/удаления сотрудника

Работа остальных справочников реализована аналогичным образом.

Следует продемонстрировать работу с документом «Приходная накладная» (см. рис. 11)

Дата
 23.10.2018
 Поставщик
 Петров П.И.
 Склад
 Основной технический
 Заведующий хозяйством
 Зеленов Х.З.

Список номенклатур

[Добавить](#)

№	Наименование номенклатуры
1	66 Samsung UE66NU7100
2	24" Samsung UE24H4080
3	ASUS X607MA-EJ066 серый
4	СКАТ Универсал 1

Редактировать Удалить Отмена

	Номер	Дата документа	Поставщик	Склад приема	Заведующий складом
✓	000000001	23.10.2018 0:00:00	Петров П.И.	Основной технический	Зеленов Х.З.
✓	000000002	14.11.2018 14:02:57	Матрико	Основной технический	Зеленов Х.З.

Сложность состоит в том, что документ «Приходная накладная» фактически состоит из двух таблиц. И для добавления элементов табличной части был реализован дополнительный элемент отображения таблиц «GridView». Функции добавления/модификации/удаления работают аналогично для всех видов таблиц (реализован один универсальный метод), однако для документов необходима реализация проведения документа. Это было реализовано с помощью POST-запроса OData с указанием метода для проведения документов «Post()».

Заключение

Выполнение данной лабораторной работы было нацелено на развитие навыков работы с клиент-серверными приложениями и взаимодействия с REST интерфейсами. В процессе разработки приложения «Складской учет» были использованы различные актуальные технологии, применяемые в разработке ПО. Среди них – ASP.NET, 1С:Предприятие, OData, RestSharp. Также следует отметить, актуальность разработки связанной с настройкой взаимодействия двух разных платформ – 1С:Предприятие и ASP.NET. Работа включала в себя реализацию двух важных аспектов функционирования информационной системы: проектирование и реализация. Разработка проекта была разделена на несколько этапов:

Первый этап – проектирование информационной системы включает в себя анализ и разработку различных спецификаций к проектируемой системе. На данном этапе были сформулированы перечни специфических особенностей проектируемой системы, а также разработана диаграмма нотации IDEF1x, отражающая основную структуру дальнейшего приложения.

Второй этап, включал в себя разработку и настройку серверной части приложения. На данном этапе был реализован REST интерфейс для взаимодействия двух разных платформ – 1С:Предприятие и ASP.NET.

Третий этап, подразумевал под собой разработку пользовательского интерфейса и программную реализацию. В рамках данного этапа, на основе разработанной на предыдущем этапе модели было разработано приложение, обладающее, необходимой в рамках поставленных задач, функциональностью.

Заключительными этапами разработки проекта являются внедрение и эксплуатация продукта. За дальнейшим развитием системы стоит тестирование, с помощью которого будут дорабатываться упущенные моменты на стадиях проектирования и разработки.