**Оглавление**

[Введение 1](#_Toc77983498)

[1. Характеристика УП «Минское отделение Белорусской железной дороги» 2](#_Toc77983499)

[2. Индивидуальное задание 4](#_Toc77983500)

[2.1 Слой данных 4](#_Toc77983501)

[2.1.1 Таблицы и их связи 4](#_Toc77983502)

[2.1.2 Хранимые процедуры 5](#_Toc77983503)

[2.2 Слой логики 7](#_Toc77983504)

[2.3 Слой клиента 9](#_Toc77983505)

[2.3.1 Авторизация 9](#_Toc77983506)

[2.3.2 Работа с обычными объектами 10](#_Toc77983507)

[2.3.3 Работа со сломанными объектами 11](#_Toc77983508)

[2.3.4 Создание отчёта списанных объектов 11](#_Toc77983509)

[2.4 Обзор генерируемого материального отчёта 12](#_Toc77983510)

[Заключение 14](#_Toc77983511)

[Список используемых источников 15](#_Toc77983512)

Введение

Технологическая практика студентов является необходимым и важнейшим этапом в системе подготовки высококвалифицированных специалистов. Она является составной частью учебного процесса и направлена на закрепление и углубление теоретических знаний, приобретение студентами навыков практической работы путем самостоятельного выполнения заданий.

Технологическая практика проводится для закрепления теоретических знаний, полученных в университете путём глубокого изучения структуры и работы технологических подразделений предприятия, где проводится практика, а также приобретение профессиональных навыков и навыков работы в трудовом коллективе.

Местом прохождения технологической практики является УП «Минское отделение Белорусской железной дороги» Молодечненская дистанция сигнализации и связи. Сроком прохождения технологической практики является период с 28 июня по 23 июля 2021 года.

Целью практики являлась разработка системы материального отчёта. В ходе практики необходимо было выполнить следующие задачи:

1. Ознакомиться со структурой и деятельностью предприятия;

2. Провести анализ, проектирование и разработку веб-приложения «Система материального отчёта».

1. Характеристика УП «Минское отделение Белорусской железной дороги»

Минское отделение Белорусской железной дороги – это 29 обособленных структурных подразделений (филиалов), 72 станции, 4 вокзала, 126 остановочных пунктов, 2790,7 км развернутой длины пути, более 16 тысяч работников.

Отделение железной дороги располагается в пределах 3 областей республики: Минской, Витебской и Гродненской, которые по своему географическому положению граничат с Российской Федерацией и Литвой. Имеет два стыковых пункта с сопредельными дорогами: Московская железная дорога — Осиновка (Красное), Литовская железная дорога — Гудогай (Кяна).



Рисунок 1.1 – Схема Минского отделения Белорусской железной дороги

Основные направления деятельности: перевозка грузов, пассажиров, багажа и грузобагажа железнодорожным транспортом общего пользования.

Ежесуточно со станций и остановочных пунктов Минского отделения железной дороги отправляется в путь более 112 тысяч человек. Именно здесь осуществляется более 40% всех пассажирских перевозок Белорусской железной дороги.

Минское отделение железной дороги — важнейшее звено в системе обеспечения международной перевозки грузов. Оно находится на пересечении двух общеевропейских транспортных коридоров: номер II (Запад-Восток) и номер IX (Север-Юг) с ответвлением IXB (Жлобин-Минск-Гудогай и далее до портов Балтийского моря: Клайпеды и Калининграда).

В зоне обслуживания Минского отделения железной дороги расположены такие крупные предприятия, как ОАО «МАЗ» — управляющая компания холдинга «Белавтомаз», РУП «Минский тракторный завод», ЗАО «Атлант», ОАО «Керамин», ОАО «БелАЗ» — управляющая компания холдинга «Белаз -ХОЛДИНГ», ОАО «Амкодор» v управляющая компания холдинга», ОАО «Завод строительных материалов» и другие.

Минское отделение железной дороги выполняет примерно треть всех грузовых перевозок Белорусской железной дороги. Более половины из них — транзитные: нефть, черные металлы, лесоматериалы, удобрения, уголь, строительные материалы и т. д.

Прохождение текущей практики происходит в Молодечненской дистанции сигнализации и связи Минского отделения Белорусской железной дороги. Организационно-функциональная структура данного отделения выглядит следующим образом (рисунок 1.2).

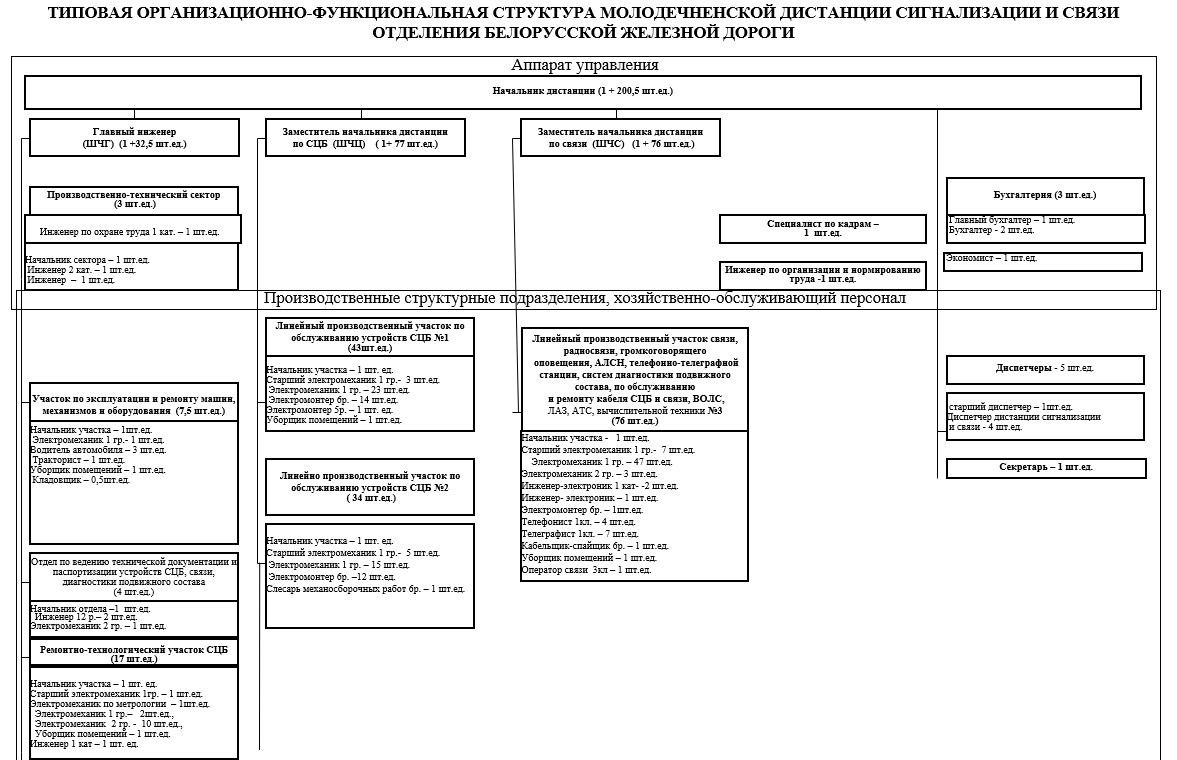


Рисунок 1.2 – Типовая структура Молодечненской дистанции сигнализации и связи

Во всяком случае, в данном отделении не производится никакого программного обеспечения для использования вне Белорусской железной дороги. Системный отдел занимается в основном установкой уже готового программного обеспечения на используемое оборудование, а также занимается поддержанием в работоспособном состоянии всех используемых систем. Разработка собственных программных решений происходит в основном по инициативе самого отдела ради обеспечения автоматизации рутинных операций.

2. Индивидуальное задание

Заданием было создание веб-приложения **«**Система материального отчёта» на основе платформы ASP.NET и модели MVC [1].

Уровни приложения:

* слой клиента;
* слой логики;
* слой данных.

Последовательно рассмотрим каждый из уровней.

2.1 Слой данных

На рисунке 2.1 отображена модель базы данных, полученная в ходе анализа предметной области и необходимого функционала. Между сущностями, представленными таблицами, также настроены связи, отражающие их взаимосвязь.

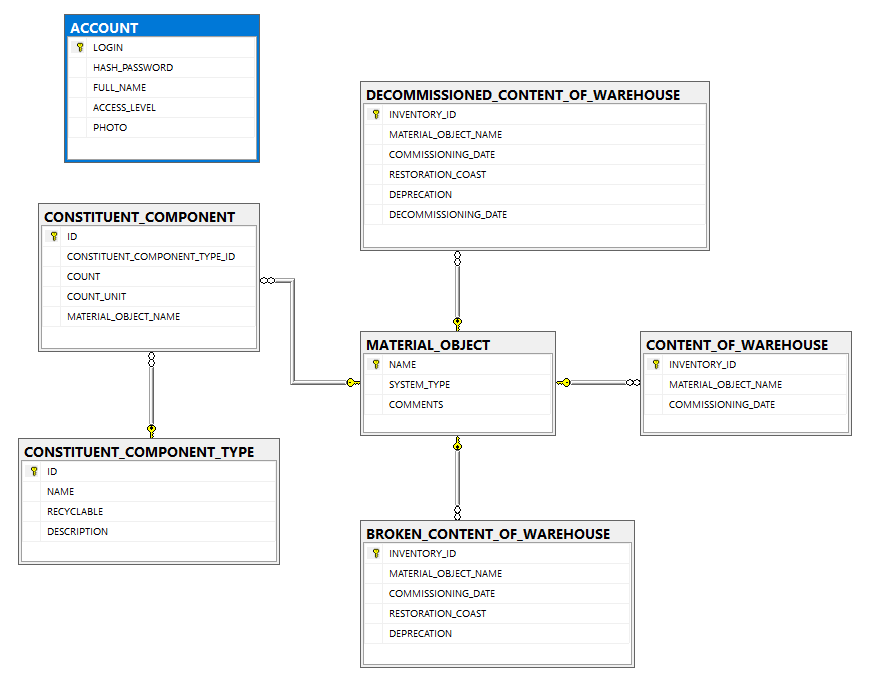


Рисунок 2.1– Структурная схема базы данных

В качестве системы управления базы данных используется Microsoft SQL [2].

2.1.1 Таблицы и их связи

На рисунке 2.1 проиллюстрированы следующие связи:

* связь «CONSTITUENT\_COMPONENT\_TYPE – CONSTITUENT\_COMPONENT»: один ко многим;
* связь «MATERIAL\_OBJECT – CONSTITUENT\_COMPONENT »: один ко многим;
* связь «MATERIAL\_OBJECT – CONTENT\_OF\_WAREHOUSE»: один ко многим;
* связь «MATERIAL\_OBJECT – BROKEN\_CONTENT\_OF\_WAREHOUSE»: один ко многим;
* связь «MATERIAL\_OBJECT – DECOMMISSIONED\_CONTENT\_OF\_WAREHOUSE»: один ко многим.

Таблица CONSTITUENT\_COMPONENT\_TYPE содержит информацию о типах компонентов (составных элементов), из которых может состоять материальный объект. Имеет поля идентификатора (ID), названия (NAME), пометка о возможности переработки элемента (RECYCLABLE) и описание (DESCRIPTION).

Таблица CONSTITUENT\_COMPONENT содержит значения количества компонентов (составных элементов), которые можно получить, при разборе материального объекта. Имеет поля идентификатора (ID), идентификатора типа компонента (CONSTITUENT\_COMPONENT\_TYPE\_ID), количество компонента (COUNT), размерность количества компонента (COUNT\_UNIT) и имя материального объекта (MATERIAL\_OBJECT\_NAME).

Таблица MATERIAL\_OBJECT содержит соответственно сами виды материальных объектов. Имеет поля имени (NAME), системного типа (SYSTEM\_TYPE) и примечания (COMMENTS).

Таблица CONTENT\_OF\_WAREHOUSE содержит материальные объекты, которые записаны на счёт складского инвентаря. Имеет поля инвентарного идентификатора (INVENTORY\_ID), имени материального объекта (MATERIAL\_OBJECT\_NAME) и даты оприходования (COMMISSIONING\_DATE).

Таблица BROKEN\_CONTENT\_OF\_WAREHOUSE также содержит материальные объекты, которые записаны на счёт складского инвентаря, но уже те, которые имеют некоторую степень изношенности и которые требуют затраты на восстановление. В дополнение к полям предыдущей таблицы добавляются поля цены восстановления (RESTORATION\_COAST) и износа (DEPRECATION).

Таблица DECOMMISSIONED\_CONTENT\_OF\_WAREHOUSE тоже содержит материальные объекты, но уже те, которые списаны со счёта складского инвентаря, ввиду нерентабельности в восстановлении материального объекта или по другой причине. В дополнение к полям предыдущей таблицы добавляется поле даты списания (DECOMMISSIONING\_DATE).

Таблица ACCOUNT содержит информацию об аккаунтах, предназначенные для входа и дальнейшего использования системы материального отчёта. Имеет поля логина (LOGIN), хэша пароля (HASH\_PASSWORD), полного имени владельца аккаунта (FULL\_NAME), уровня доступа (ACCESS\_LEVEL) и фотографии пользователя (PHOTO).

2.1.2 Хранимые процедуры

Для взаимодействия с данными используются преимущественно процедуры.

Процедуры для работы с аккаунтами:

* CreateAccount – добавление записи в таблицу аккаунтов;
* Authenticate – аутентификация аккаунта;
* GetUserInfoByLogin – получение информации аккаунта, без хэша пароля.

Пример процедуры указан на рисунке 2.2.



Рисунок 2.2 – Пример разработанной хранимой процедуры

Процедуры для работы с компонентами и материальными объектами:

* GetMaterialObjectByName – получение материального объекта по имени;
* GetConstituentComponentsByMaterialObjectName – получение количеств компонент, из которых состоит материальный объект под именем;
* GetConstituentComponentTypesByMaterialObjectName – получение типов компонент, из которых состоит материальный объект под именем;
* GetConstituentComponentTypeById – получение типа компонента под указанным идентификатором.

Процедуры для работы со складом:

* AddContentToWarehouse – добавление записи в таблицу инвентаря склада;
* GetContentFromWarehouseById – получить элемент инвентаря склада по указанному идентификатору;
* GetBrokenContentFromWarehouseById – получить повреждённый элемент инвентаря склада по указанному идентификатору;
* GetDecommissionedContentFromWarehouseById – получить списанный элемент инвентаря склада по указанному идентификатору;
* GetContentOfWarehouseByIdRange – получить диапазон из 20 элементов инвентаря склада по указанному номеру диапазона;
* GetContentPartsCount – получить количество диапазонов по 20 элементов инвентаря склада;
* GetBrokenContentOfWarehouseByIdRange – получить диапазон из 20 поврежденных элементов инвентаря склада по номеру диапазона;
* GetBrokenContentPartsCount – получить количество диапазонов по 20 сломанных элементов инвентаря склада;
* GetDecommissionedContentOfWarehouseByIdRange – получить диапазон из 20 списанных элементов инвентаря склада по номеру диапазона;
* GetDecommissionedContentPartsCount – получить количество диапазонов по 20 списанных элементов инвентаря склада;
* FromSimpleToBroken – перенос элемента из обычного инвентаря в инвентарь повреждённых элементов;
* FromBrokenToSimple – перенос элемента из повреждённого инвентаря в инвентарь обычных элементов;
* FromBrokenToDecommissioned – перенос элемента из повреждённого инвентаря в инвентарь списанных элементов.

2.2 Слой логики

Для взаимодействия с базой данных используется технология ADO.NET [3] в связке с ORM Dapper [4]. Для этого были разработаны классы:

* MRConnection – класс, предназначенный для получения подключения к базе данных SqlConnection;
* AuthenticationProvider – класс, предназначенный для работы с аккаунтами с помощью подключения к базе данных SqlConnection;
* MaterialsAndComponentsProvider – класс, предназначенный для работы с материальными объектами и компонентами с помощью подключения к базе данных SqlConnection;
* WarehouseProvider – класс, предназначенный для работы со складским инвентарём с помощью подключения к базе данных SqlConnection.

Пример провайдера, класса для взаимодействия с процедурами базы данных указан на рисунке 2.3.

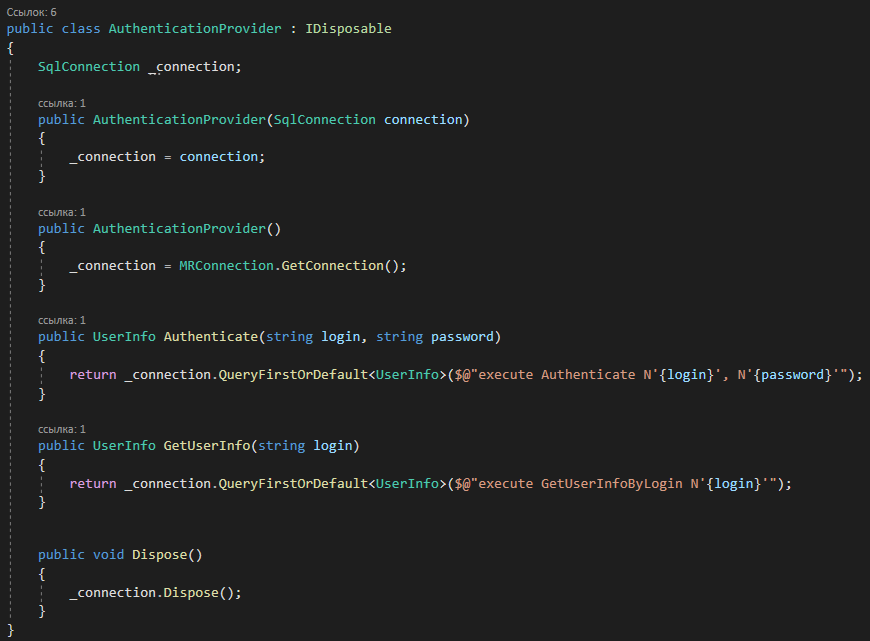


Рисунок 2.3 – Пример разработанного провайдера

Были разработаны классы-модели для представления записей из таблиц базы данных, описанной ранее:

* LoginModel – класс для представления данных для входа в систему;
* UserInfo – класс, содержащий данные ACCOUNT, исключая поле хэша пароля (HASH\_PASSWORD);
* MaterialObject – класс, который представляет запись таблицы MATERIAL\_OBJECT;
* ConstituentComponentType – класс, который представляет запись таблицы CONSTITUENT\_COMPONENT\_TYPE;
* ConstituentComponent – класс, который представляет запись таблицы CONSTITUENT\_COMPONENT;
* ContentOfWarehouse – класс, который представляет запись таблицы CONTENT\_OF\_WAREHOUSE;
* BrokenContentOfWarehouse – класс, который представляет запись таблицы BROKEN\_CONTENT\_OF\_WAREHOUSE;
* DecommissionedContentOfwarehouse – класс, который представляет запись таблицы DECOMMISSIONED\_CONTENT\_OF\_WAREHOUSE.

Для работы с Excel-фалами используется библиотека EPPlus. Для более удобной работы были разработаны следующие структуры и классы:

* EpplusExtensions – класс, содержащий методы расширения;
* CellConfig – структура, содержащий набор значений параметров;
* ExcelCellConfiguring – класс, содержащий методы для настройки ячеек Excel-файла без или с помощью описанной выше структуры.

Для формирования отчёта были разработаны ещё два класса:

* ReportPackage – класс, собирающий информацию с базы данных, которая требуется для формирования отчёта. Класс продемонстрирован на рисунке 2.4;
* ReportPackageField – класс, содержащий данные для заполнения строки Excel-отчёта;
* ReportExcelFile – класс, содержащий методы, позволяющие формировать Excel-отчёт.

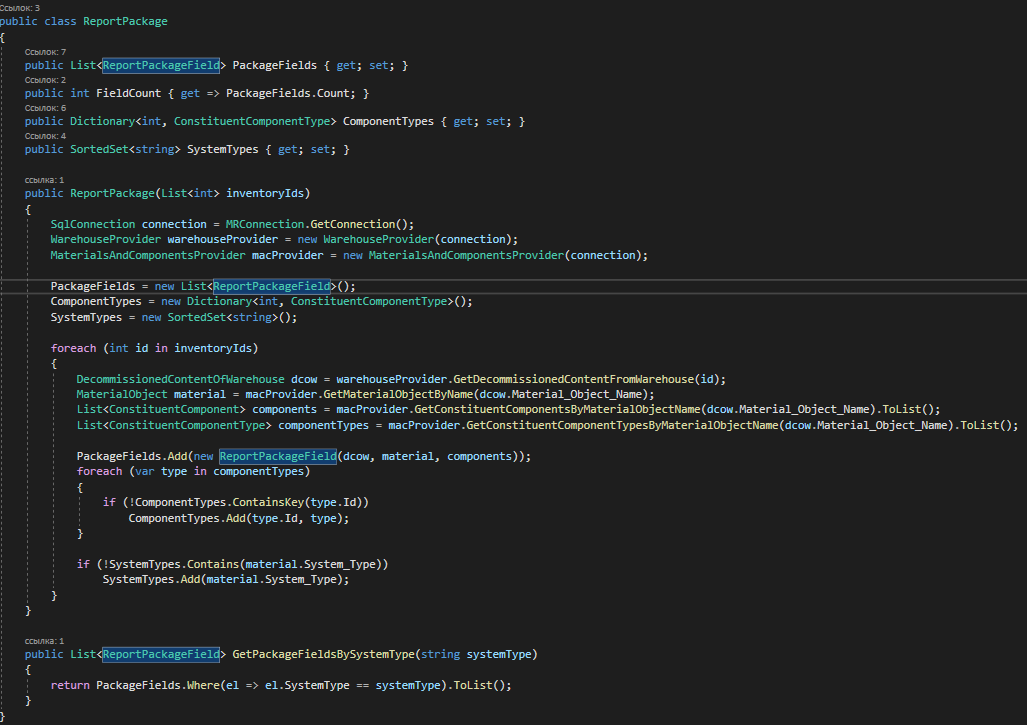


Рисунок 2.4 – Класс ReportPackage

Дополнительно был разработан класс ImageHelper, предназначенный для конвертации изображения в массив байт и обратно.

2.3 Слой клиента

Для удобного взаимодействия с системой материального отчёта был разработан интерфейс веб-приложения (рисунок 2.5).

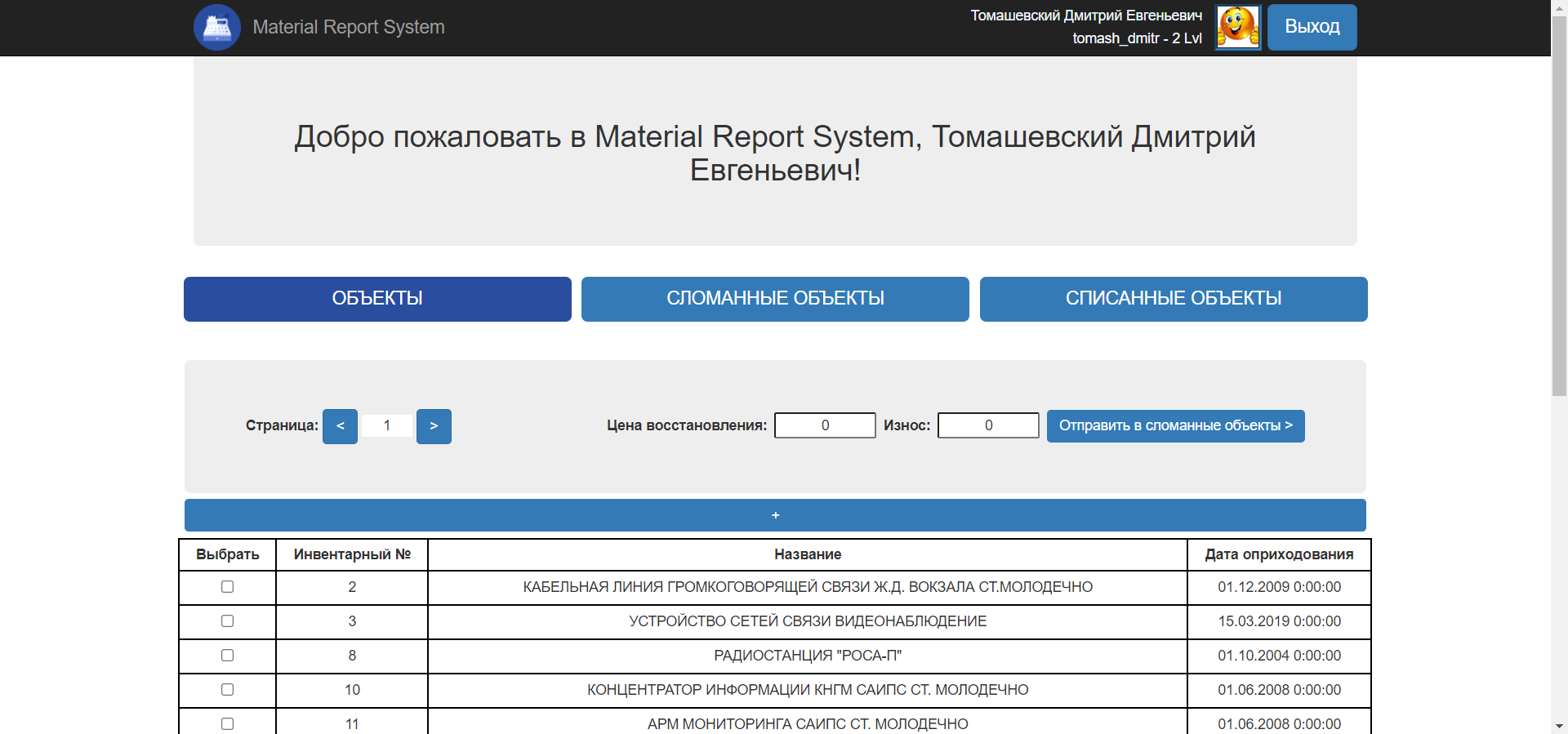


Рисунок 2.5 – Интерфейс системы материального отчёта

Далее распишем каждый момент работы с интерфейсом веб-приложения.

2.3.1 Авторизация

При первом входе в веб-приложение, будет предложено зайти в собственный аккаунт, в ином случае, дальнейшая работа будет невозможна (рисунок 2.6).

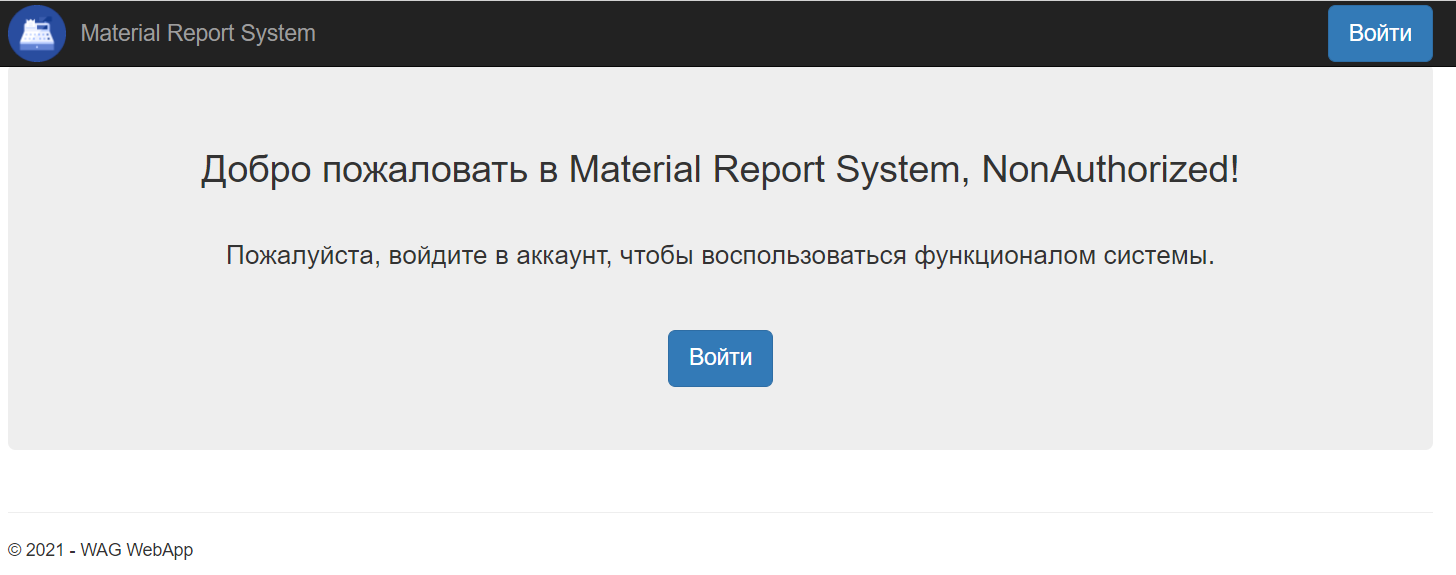


Рисунок 2.6 – Интерфейс системы без авторизации

Нажимаем любую из кнопок «Войти». Вводим свои логин и пароль и нажимаем «Вход» (рисунок 2.7).

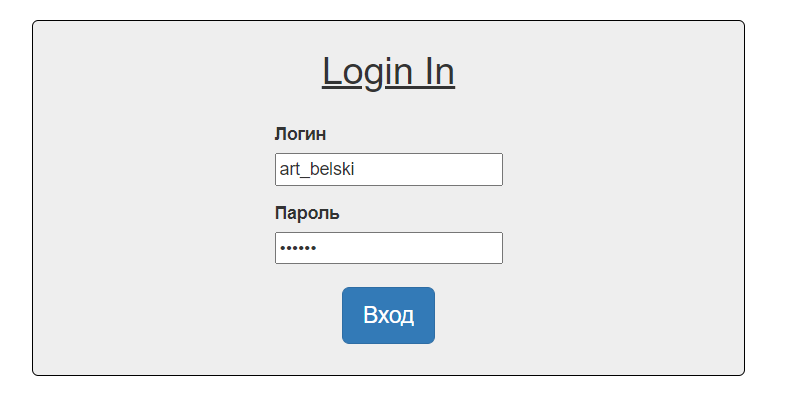


Рисунок 2.7 – Форма авторизации

Далее мы будем наблюдать сайт следующего вида с приветствием и отображаемой в правом краю информацией аккаунта (рисунок 2.8).

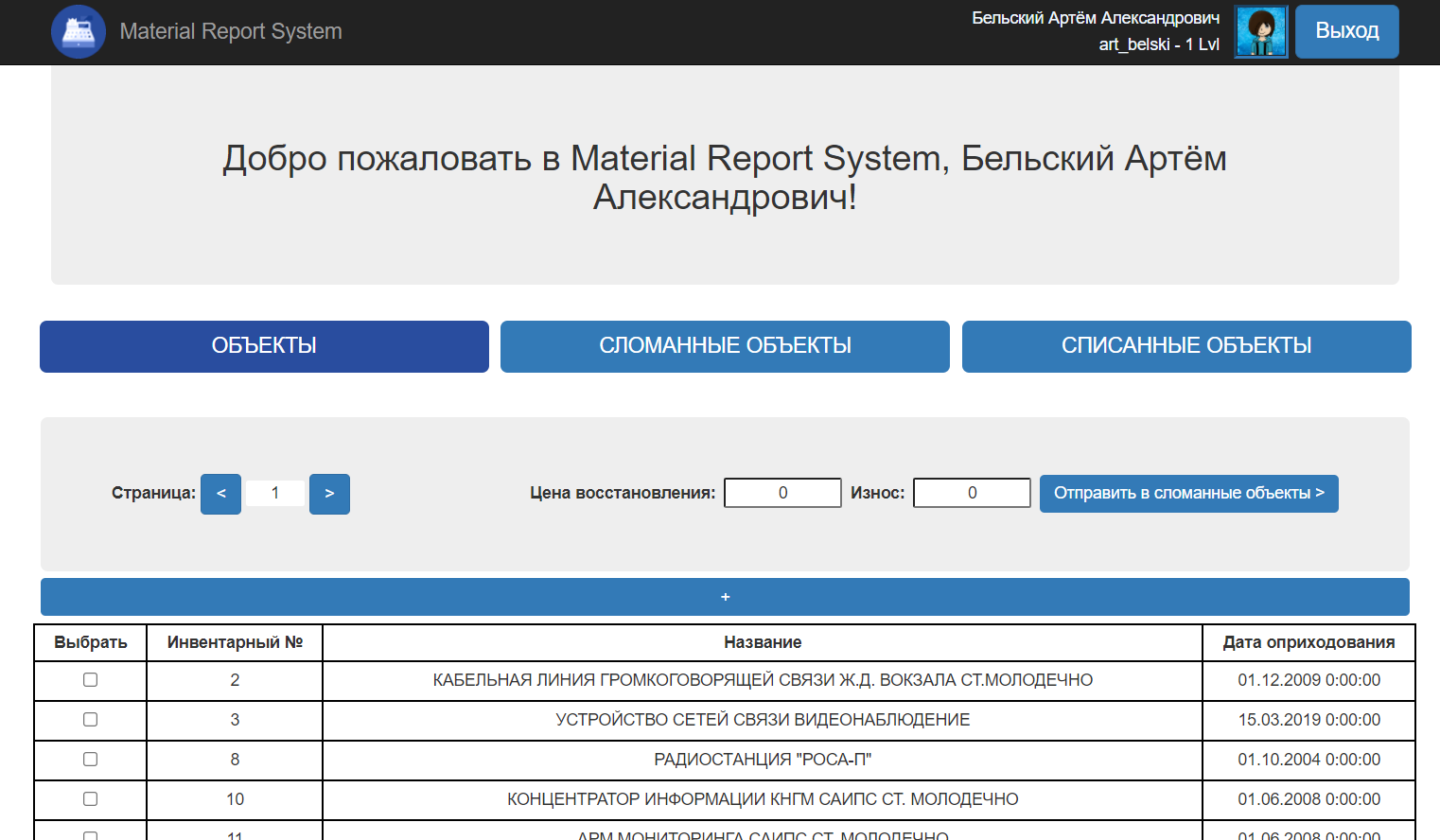


Рисунок 2.8 – Страница после входа

После проделанных манипуляций можно совершать действия с базой данных.

2.3.2 Работа с обычными объектами

В таблице ниже можно выбирать нужные объекты, находящиеся на складе (рисунок 2.9). Светлое выделение – объект выбран, тёмное – наведение на объект.



Рисунок 2.9 – Таблица с выбранными элементами

Выше таблицы находится блок управления (рисунок 2.10), где можно переключать страницы списка элементов. Правее располагаются 2 поля ввода и кнопка «Отправить в сломанные объекты >», после ввода и применения которых выбранные элементы будут переведены в таблицу сломанных объектов (BROKEN\_CONTENT\_OF\_WAREHOUSE) с полями, заполненными указанными величинами.

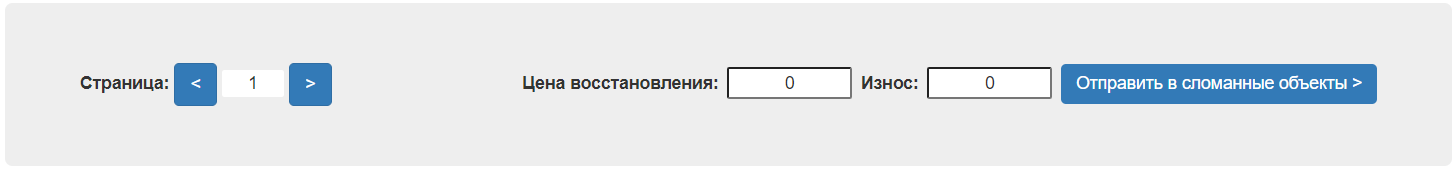


Рисунок 2.10 – Блок управления вкладки «Объекты»

Ниже, при нажатии на кнопку «+» раскроется область с формой для добавления нового объекта в инвентарь склада (рисунок 2.11).

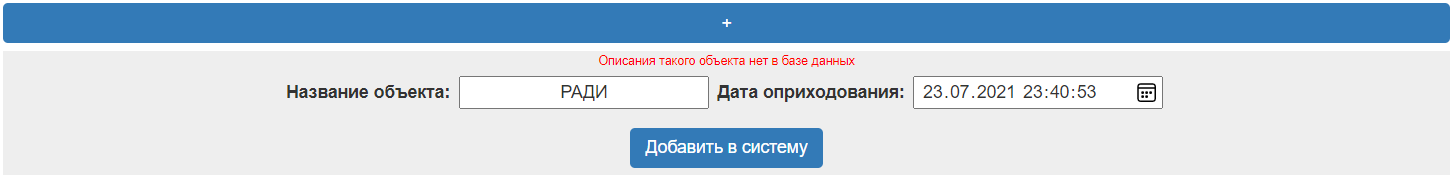


Рисунок 2.11 – Область формы добавления объекта

При вводе в поле «Название объекта» некорректного названия, будет высвечиваться предупреждение, что такого объекта не существует. При нажатии «Добавить в систему» объект добавляется в таблицу CONTENT\_OF\_WAREHOUSE.

2.3.3 Работа со сломанными объектами

Главное отличие от предыдущей вкладки, во вкладке «Сломанные объекты» блок управления выглядит несколько иначе (рисунок 2.12).

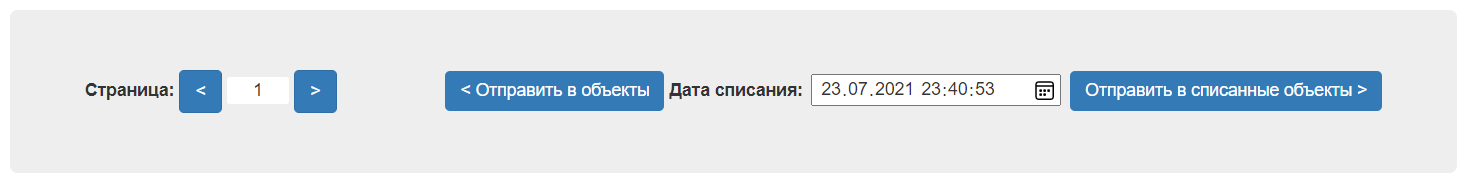


Рисунок 2.12 – Блок управления вкладки «Сломанные объекты»

Кнопка «< Отправить в объекты» возвращает выбранные объекты обратно в таблицу CONTENT\_OF\_WAREHOUSE. При заполнении области ввода «Дата списания» и нажатии на кнопку «Оправить в списанные объекты >» помещает выбранные элементы в таблицу DECOMMISSIONED\_CONTENT\_OF\_WAREHOUSE.

2.3.4 Создание отчёта списанных объектов

Как и вкладка «Сломанные объекты», вкладка «Списанные объекты» блок управления также выглядит несколько иначе (рисунок 2.13).

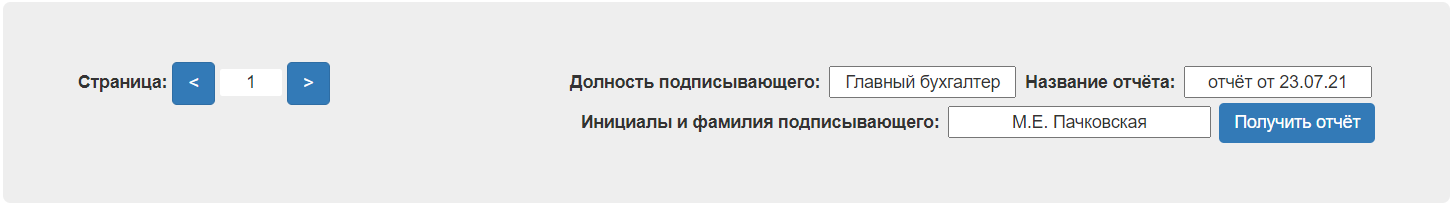


Рисунок 2.13 – Блок управления вкладки «Списанные объекты»

После заполнения областей ввода «Должность подписывающего», «Название отчёта» и «Инициалы и фамилия подписывающего» и нажатии на кнопку «Получить отчёт» появляется запрос от браузера на загрузку и сохранение Excel-файла с отчётом по выбранным элементам.

2.4 Обзор генерируемого материального отчёта

Полученный отчёт состоит из, как минимум, 3 вкладок. Количество дополнительных вкладок зависти от количества системных типов выбранных элементов. Первая вкладка «Основные» представляет собой список из выбранных элементов с добавлением области с указанной должностью, место подписи и ФИО подписывающего (рисунок 2.14). Эта вкладка, подразумевается, должно использоваться как форма для печати.

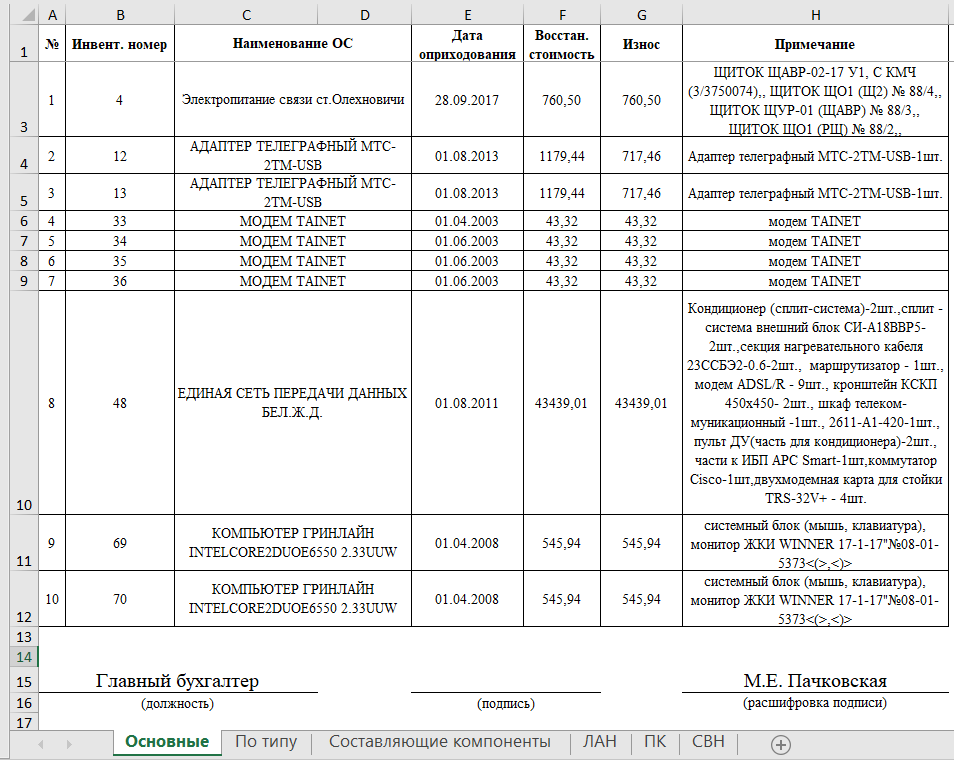


Рисунок 2.14 – Вкладка «Основные»

Вторая вкладка «По типу» представляет собой просто таблицу с элементами с возможностью отфильтровать таблицу по колонкам (рисунок 2.15).

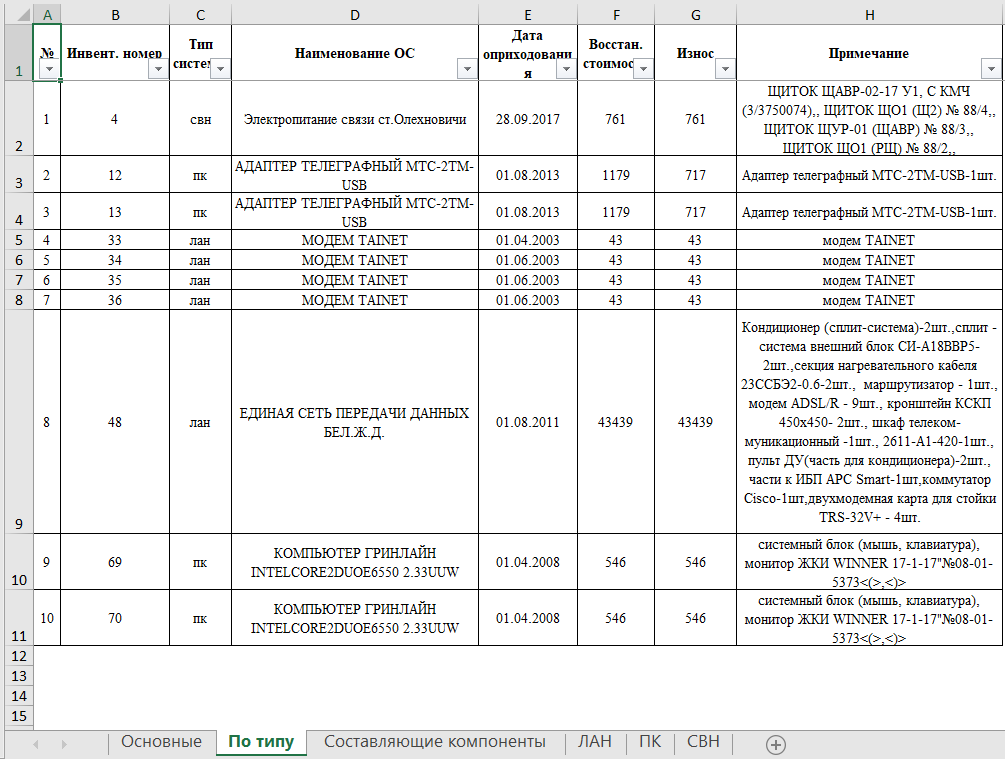


Рисунок 2.15 – Вкладка «По типу»

Третья вкладка «Составляющие компоненты» имеет список с дополнительными колонками, в которых располагаются значения количества составляющих элементов. Жёлтым цветом помечена строка, которая рассчитывает сумму каждого из составляющих материальных объектов (рисунок 2.16). На странице также можно использовать фильтр по столбцам.

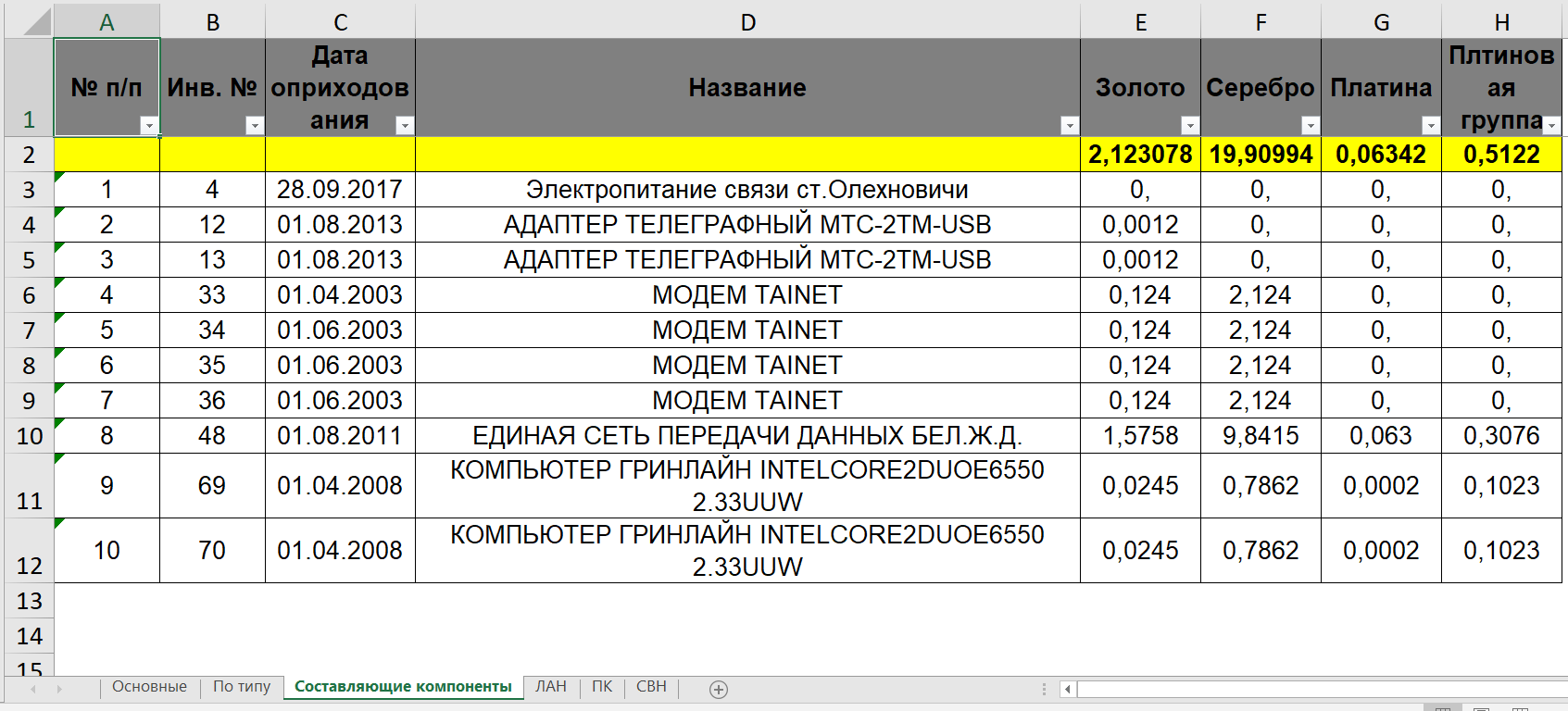


Рисунок 2.16 – Вкладка «Составляющие компоненты»

Следующие вкладки состоят исключительно из элементов определённых системных типов (рисунок 2.17).

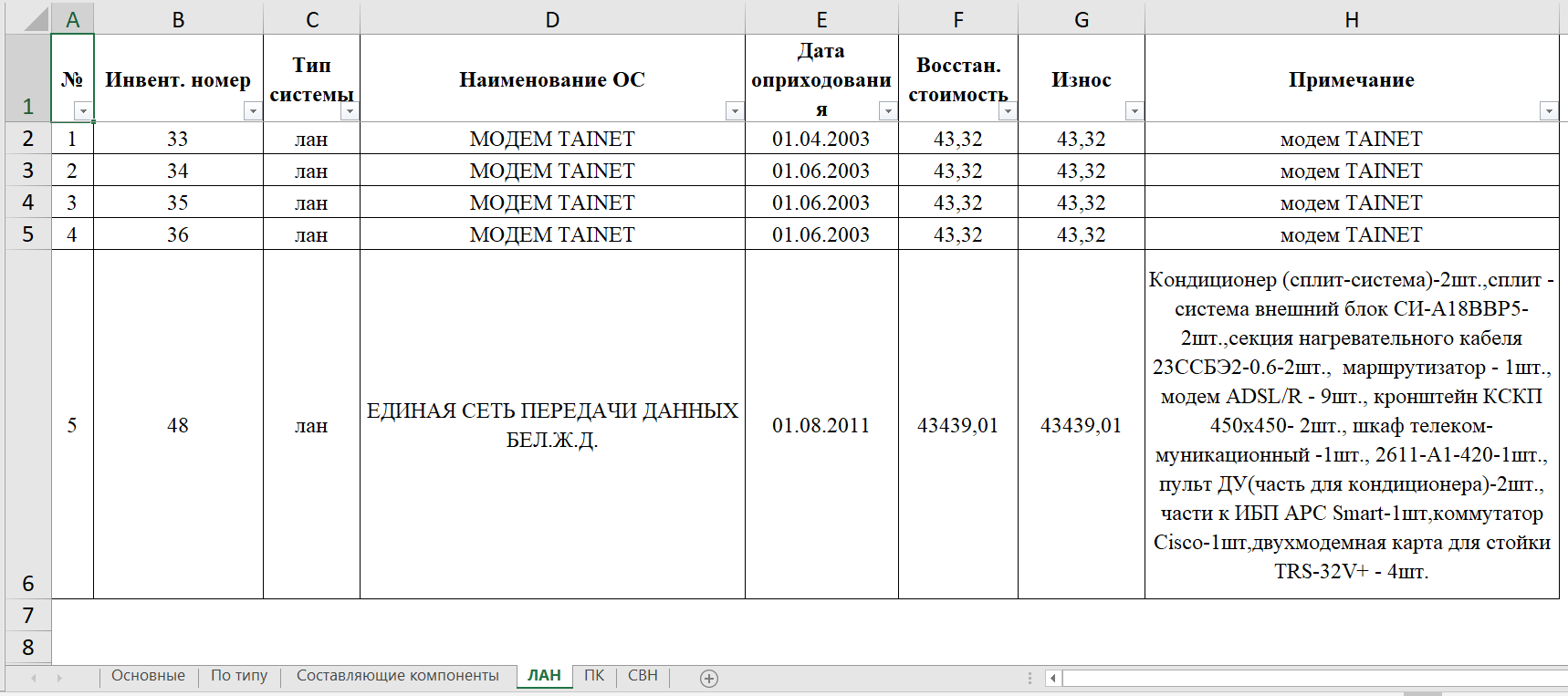


Рисунок 2.17 – Вкладка системного типа «ЛАН»

Фильтр по столбцам также присутствует.

Заключение

За время прохождения практики были изучены организационная структура предприятия и принципы работы системного отдела.

Были лучше изучены и применены на практике такие технологии как ASP.NET, ADO.NET, ORM Dapper для разработки системы материального отчёта.

Производственно-технологическая практика позволила закрепить знания, полученные на протяжении обучения в университете, а также получить новые, используемые во многих IT сферах.

Список используемых источников

1. Сайт metanit.net – Электронный ресурc. – Ссылка: [ASP.NET MVC 5 | Полное руководство](https://metanit.com/sharp/mvc5/).
2. Сайт metanit.net – Электронный ресурс. – Ссылка: [Microsoft SQL Server 2019 и T-SQL](https://metanit.com/sql/sqlserver/).
3. Сайт metanit.net – Электронный ресурс. – Ссылка: [Руководство по ADO.NET и работе с базами данных](https://metanit.com/sharp/adonet/).
4. Сайт dapper-tutorial.net – Электронный ресурс. – Ссылка: [Dapper Dapper Tutorial | Dapper Tutorial and Documentation](https://dapper-tutorial.net/)