**Содержание**

[Введение 2](#_Toc132625098)

[1 Общая характеристика предприятия 3](#_Toc132625099)

[2 Постановка цели и формулировка задач 5](#_Toc132625100)

[3 Анализ и сравнительный обзор аналогов 6](#_Toc132625101)

[3.1 NetSuite WMS 7](#_Toc132625102)

[3.2 Sortly 8](#_Toc132625103)

[3.3 Maropost 9](#_Toc132625104)

[3.4 Вывод по разделу 10](#_Toc132625105)

[4 Проектирование программного обеспечения 11](#_Toc132625106)

[4.1 Проектирование базы данных 11](#_Toc132625107)

[4.2 Архитектура веб-приложения 12](#_Toc132625108)

[4.3 Архитектура серверной части веб-приложения 13](#_Toc132625109)

[4.4 Архитектура клиентской части веб-приложения 14](#_Toc132625110)

[4.5 Проектирование пользовательского интерфейса 15](#_Toc132625111)

[4.6 Вывод по разделу 16](#_Toc132625112)

[Заключение 17](#_Toc132625113)

[Список использованных источников 18](#_Toc132625114)

# **Введение**

Данный отчёт содержит описание проведённой преддипломной практики.

Местом прохождения практики является ЗАО «Итранзишэн».

В цели практики входит:

* получение практических навыков по специальности;
* закрепление знаний, полученных на лекциях и лабораторных работах;
* подготовка к выполнению выпускной квалификационной работы.

В задачи практики входит:

* изучение стандарта БГТУ по дипломному проектированию;
* проведение анализа информационных источников (системы, методы, алгоритмы, программные решения, технологии программирования) по теме дипломного проекта в соответствии с приказом №72-С от 13.03.2023г.;
* проведение анализа требований, разработка функциональных требований и архитектуры проектируемого приложения;
* выполнение реализации приложения по теме дипломного проекта в соответствии с приказом №72-с от 13.03.2023г.;
* составление отчёта о преддипломной практике.

Поставленные задачи должны быть выполнены в полной мере и в соответствии с установленным календарным графиком.

# **1 Общая характеристика предприятия**

ЗАО «Итранзишэн» – международный поставщик ИТ-решений. Компания предоставляет полный комплекс услуг по разработке ПО, интеграции приложений, ИТ-консалтинга, тестирования и поддержки ПО.

В Беларуси офисы расположены в Минске, Гродно, Могилёве, Полоцке и Бресте.

Компания основана в 1998г. в основе холдинга BelHard. В 2021 году компания насчитывала более чем 2500 сотрудников.

Структура предприятия представлена на рисунке 1.1.

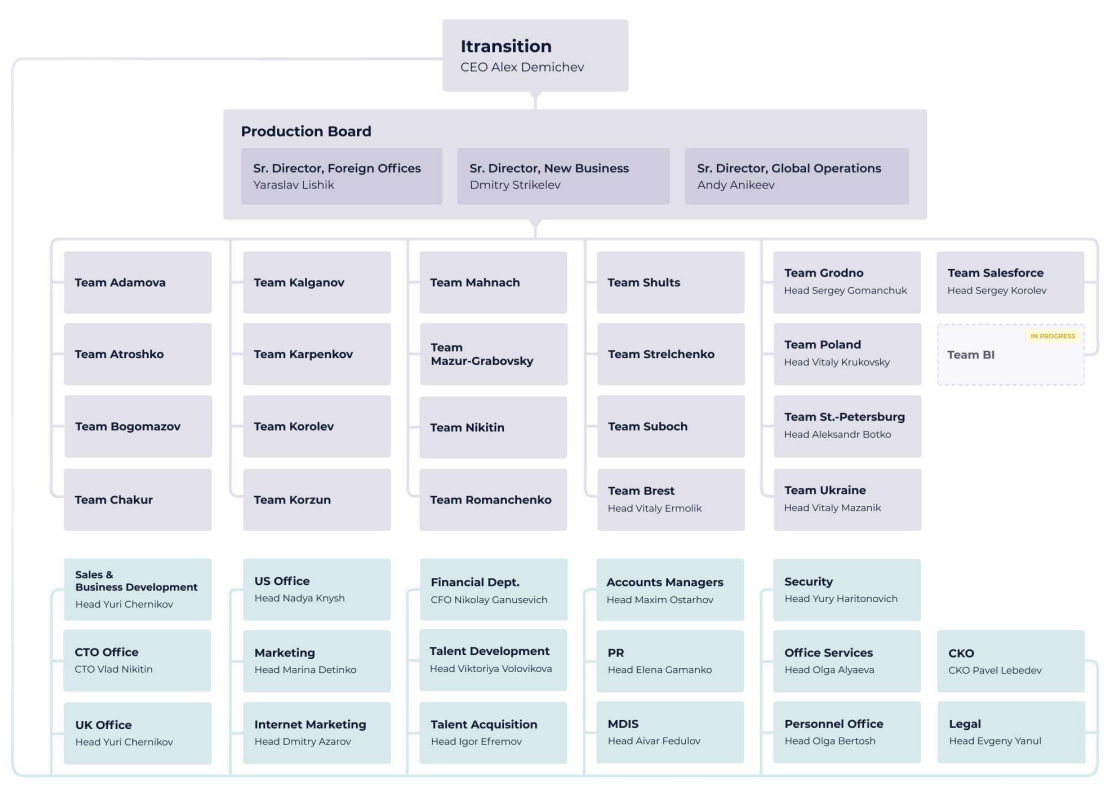


Рисунок 1.1 – Структура предприятия

В компании существуют производственные и непроизводственные подразделения.

Производство возглавляет Production Board и насчитывает три направления.

Ярослав Лишик отвечает за курирование и открытие новых офисов. Андрей Аникеев отвечает за целевые KPI производственных подразделений, Дмитрий Стикелев – за старт новых проектов. Есть 22 производственных отделов.

Также насчитывается 17 непроизводственных отделов (на схеме рисунка 1 обозначены зелёным цветом).

В ключевые роли входят:

* юнит-менеджер – прямой руководитель, отвечает за проектную ротацию, загруженность на проекте, помогает профессионально расти и развиваться;
* проектный менеджер – руководитель конкретного проекта, ведёт переговоры с клиентом, отвечает за проектную загрузку;
* ментор – куратор, который вводит новых членов проекта в курс дел, помогает разбираться в технических вопросах;
* hr – сопровождает с первого дня работы, помогает адаптироваться и узнать процессы компании, занимается мотивацией и обучением.

У предприятия есть схема карьерного роста, она представлена на рисунке 1.2.

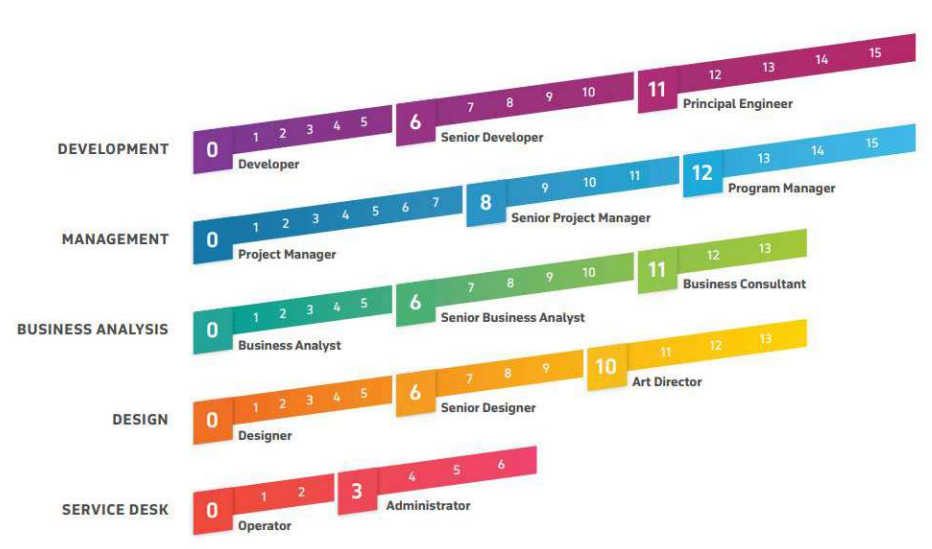


Рисунок 1.2 – Схема карьерного роста предприятия

Для каждого уровня есть разработанные требования, которые может увидеть каждый из сотрудников предприятия.

# **2 Постановка цели и формулировка задач**

Управление складом – это множество бизнес-процессов, таких как определение местоположения товара, приём и размещение новых товаров на складе, инвентаризация, контролирование работы сотрудников склада и многих других.

Большинство таких процессов являются рутинными, имеют определённый алгоритм действий. Чем больше склад, тем больше времени, а соответственно денег, они занимают. Современные технологии позволяют автоматизировать такие процессы. Так как в современном мире почти у каждого есть персональный компьютер или смартфон, где по умолчанию установлен веб-браузер, то хорошим и доступным решением является веб-приложение.

Таким образом, целью дипломного проекта является разработка веб-приложения для управления складом.

Для достижения поставленной цели сформулированы следующие задачи:

* провести анализ информационных источников;
* выбрать информационные технологии (языки программирования, платформы, алгоритмы, методы) для разработки;
* спроектировать структуру базы данных;
* спроектировать архитектуру веб-приложения;
* разработать REST API, который будет включать методы доступа для: товаров, пользователей, задач, комментариев, аутентификации, клиентов, словарей, адресов;
* разработать сервис для отправки уведомлений об изменении статуса задачи, регистрации пользователя, приближающейся просрочки выполнения задач;
* продумать и разработать дизайн пользовательского интерфейса;
* разработать клиентское приложение, которое будет использовать REST API, которое будет включать модули для: товаров; задач; словарей; адресов; панели администратора для управления пользователями и клиентами; аутентификации;
* протестировать веб-приложение;
* написать руководство пользователя;
* написать руководство по развёртыванию.

# **3 Анализ и сравнительный обзор аналогов**

Система управления складом – это программа, предназначенная для автоматизации управления процессами склада и работы складского комплекса.

Система управления складом интегрируется с учётными программами предприятия, обеспечивает полную прозрачность данных в цепочке поставок всей компании.

В основные бизнес-процессы склада входят: приёмка товара, учёт, товара, оптимизация процесса хранения, управление персоналом.

Учёт товара. Система для склада ведёт учёт товара, предоставляет пользователям актуальную информацию о его количестве, позволяет минимизировать затаривание склада ненужными остатками. Экономия складского пространства и бесперебойная работа логистики склада достигается за счёт оптимального распределения и хранения запасов.

Оптимизация процесса хранения. Система предоставляет пользователям возможность моделирования эффективных схем хранения различных товаров, учитывающих их характеристики, например, такие как, вес товара или его спрос (скорость оборачиваемости товарных запасов). Это позволяет организовать процесс хранения таким образом, что более востребованные или тяжёлые товары будут располагаться ближе к зоне отгрузки или товар, отгружаемый вместе, будет храниться рядом друг с другом, за счёт чего времени для его обработки потребуется меньше. Учёт многочисленных факторов хранения обеспечивает эффективную работу складской логистики.

Управление персоналом. Централизованное управление складом посредством складской системы сокращается необходимость в содержании персонала в большом количестве. Оптимизация рабочего фонда становиться возможной, в том числе, за счёт сокращения частоты инвентаризации товара. Программа позволяет производить инвентаризацию товара без вмешательства в повседневную работу склада. Сокращение расходов на оплату труда позволяет снизить текущие (операционные) расходы на содержание склада и повысить эффективность работы всего предприятия. Измерение ключевых показателей эффективности работы склада повышает эффективность работы, позволяет измерять показатели эффективности, проверять выполнение и результативность работы, формировать форму отчётности, настраивать систему мотивации и нормы оплаты труда.

Документооборот. Система управления складом способна автоматизировать все процессы, устраняя необходимость ведения бумажного документооборота, требующего значительных ресурсов. Пользователи могут иметь общий доступ к базе данных, обеспечить работников необходимой информацией для быстрой и качественной работы.

Управление складом и контроль. Для предприятий, которым необходимы расширенные возможности контроля, программа предлагает отслеживать товар по различным характеристикам: серийные номера, сроки годности, товарные коды и т.п.

На рынке представлены различные типы систем управления складом, есть простые коробочные решения, а есть сложные комплексные системы управления складом, подходящие для крупных складов.

## **3.1 NetSuite WMS**

Одним лучших примеров системы управления складом является продукт от Oracle – NetSuite WMS [1] (рисунок 3.1).

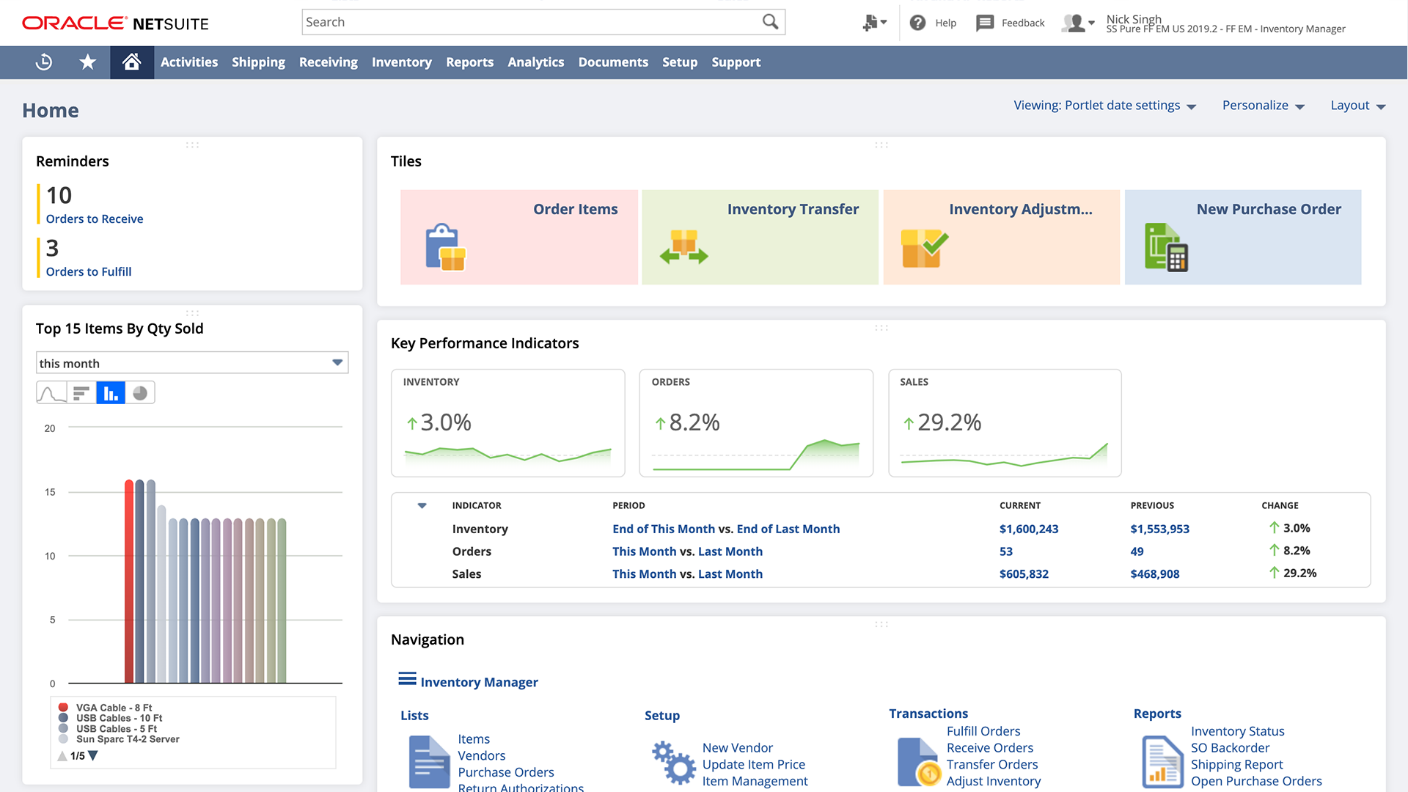


Рисунок 3.1 – NetSuite WMS

Информационная система помогает оптимизировать складские и производственные операции. Это помогает пользователям выполнять основные функции склада, такие как получение, хранение и отгрузка товаров.

Особенности инструмента:

* управление задачами;
* мобильное сканирование штрих-кода;
* квитанция об авторизации;
* определение стратегии складывания и сбора;
* данные в реальном времени, управление заказами, отслеживание штрих-кода, выставление счетов, управление рабочей силой и нагрузкой и т.д.

Плюсы:

* простой и удобный интерфейс;
* надёжная компания (Oracle);
* сторонние плагины;
* беспроводное складирование;

К минусам можно отнести высокую цену и неэффективность при использовании в качестве ERP.

## **3.2 Sortly**

Sortly [2], несомненно, является одним из лучших визуальных программ для управления складом на основе фотографий для малого бизнеса. С этим программным обеспечением вы получаете полностью настраиваемую систему, которая может отслеживать любой предмет или относящуюся к нему информацию в разных местах.

Пользователи Sortly могут добавлять настраиваемые поля, квитанции и несколько изображений к каждому элементу, чтобы упростить отслеживание активов, независимо от того, на каком складе находится товар. Пользователи также могут создавать и печатать QR-этикетки и штрих-коды, которые можно сканировать с помощью Sortly.

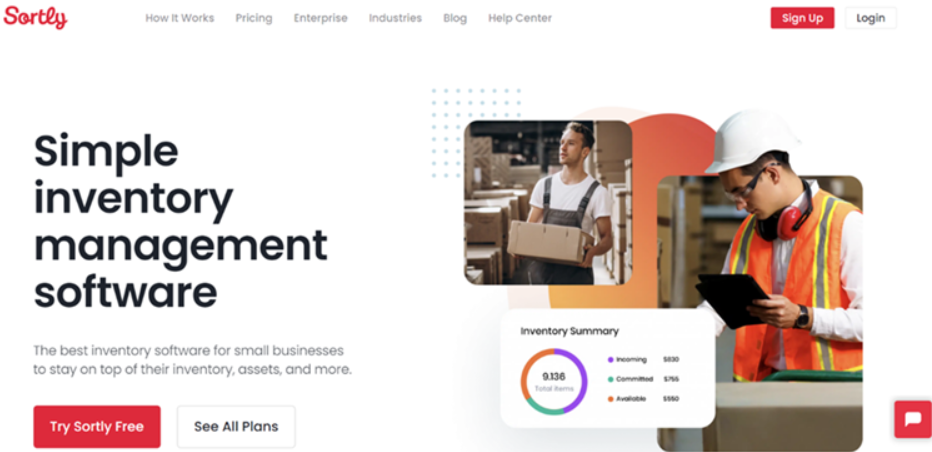


Рисунок 3.2 – Sortly

Особенности инструмента:

* сканирование и обновление товаров с помощью QR-меток и штрих-кодов;
* создание автоматического оповещения для отслеживания уровня запасов;
* назначение ролей пользователей и управление правами доступа;
* отслеживание инвентаря и активности пользователей;
* создание собственных отчётов в формате CSV и PDF.

Плюсы:

* прост в использовании, никаких технических знаний не требуется;
* кроссплатформенное программное обеспечение, которое работает практически на всех типах устройств;
* чистая и всеобъемлющая пользовательская панель;
* легко создаваемые персонализированные QR-метки и штрих-коды;
* доступна бесплатная версия.

Минусы:

* нет интеграций;
* будет предоставляться только справочный центр и поддержка по электронной почте, если не купить индивидуальный план.

## **3.3 Maropost**

Maropost [3] — это платформа, которая может не только автоматизировать процесс управления запасами, но и упростить управление вашими заказами. Платформа отличается от своих конкурентов тем, что упрощает централизованное управление несколькими интернет-магазинами.

Проще говоря, вы можете положиться на эту платформу для управления несколькими магазинами с единой панели управления. По сути, вы видите всю свою империю онлайн-рынка с высоты птичьего полета. Вы можете легко управлять всеми списками продуктов, заказами и информацией, относящейся к клиентам, с единой интуитивно понятной платформы.

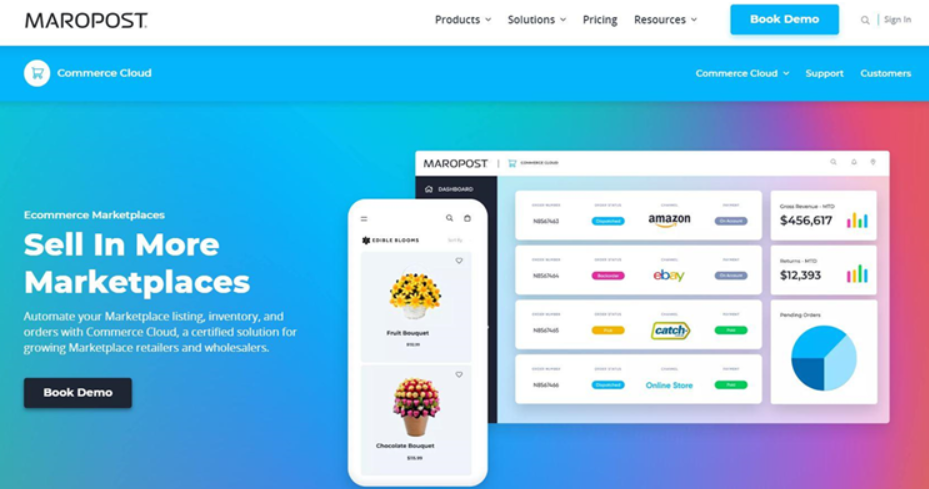


Рисунок 3.3 – Maropost

Особенности инструмента:

* управление несколькими каналами продаж из одного месте;
* создание индивидуального интернет-магазина;
* оптимизация управления клиентами;
* установка правил ценообразования;
* синхронизация данных отслеживания доставки.

Плюсы:

* удобный интерфейс;
* централизированное решение для нескольких функций, связанных с электронной коммерцией;
* подробный список каналов и инвентаря;
* гибкое ценообразование.

Минусы:

* может не подходить для малого бизнеса.

## **3.4 Вывод по разделу**

В результате исследования существующих аналогов приложений для управления складом были определены наиболее востребованные функции, плюсы и минусы подобных систем.

Полученные результаты исследования будут учтены при разработке, а также использованы для оценки и определения сложных мест программы.

# **4 Проектирование программного обеспечения**

## **4.1 Проектирование базы данных**

Для хранения данных в приложении используется реляционная база данных MS SQL Server.

Логическая схема базы данных, спроектированной в ходе разработки приложения, представлена на рисунке 4.1.

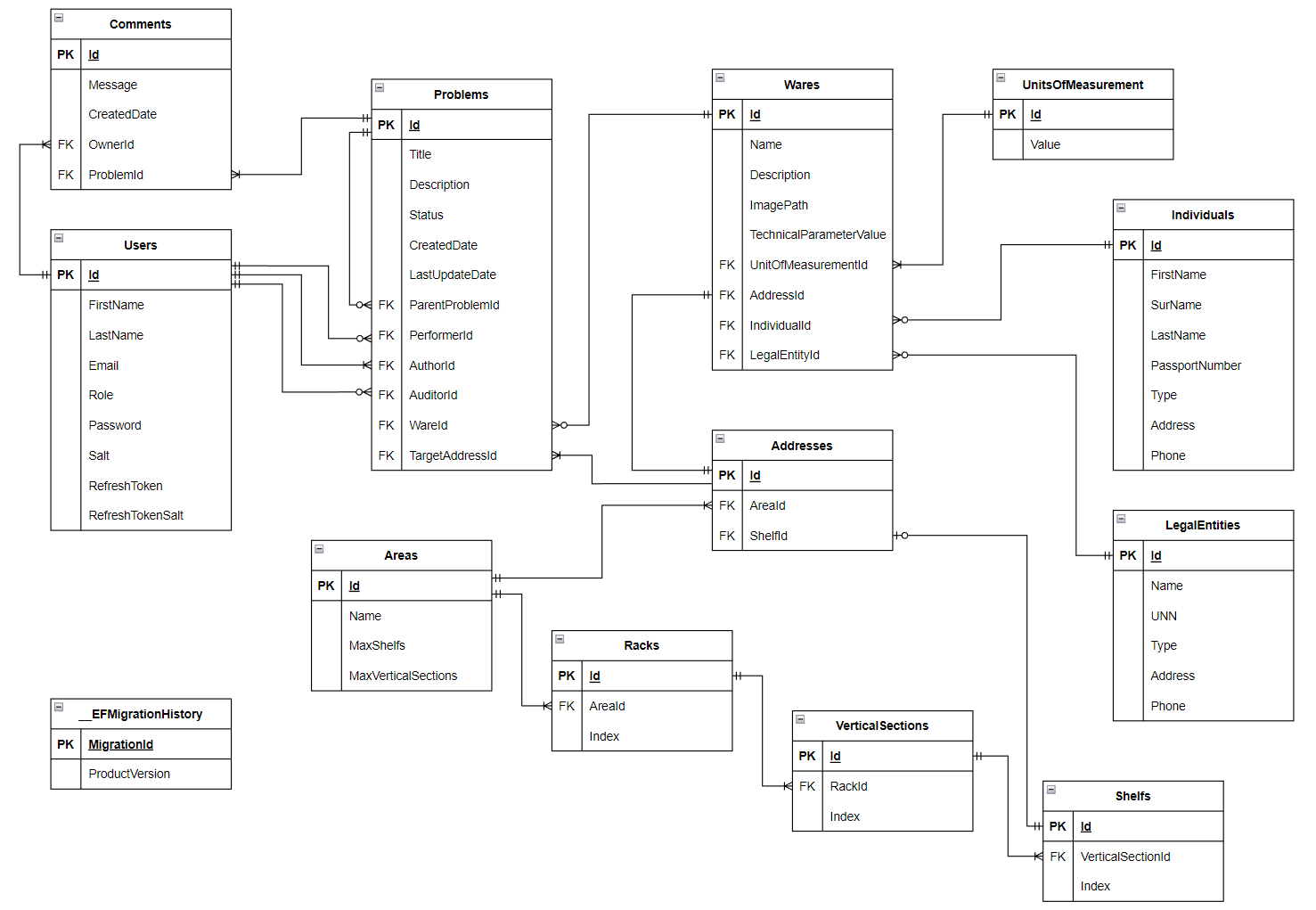


Рисунок 4.1 – Диаграмма базы данных

База данных состоит из 12 основных таблиц (бизнес сущностей), таблицы содержащей историю миграций и 11 таблиц Hangfire.

Основными сущностями в базе данных являются Users, Problems и Wares.

Таблица Users содержит данные о сотрудниках склада, именно они будут в основном взаимодействовать с системой.

Таблица Wares содержит данные товаров, которые находятся на складе.

Таблица Problems содержит информацию о задачах, это могут независимы от товаров задачи или наоборот связанными с ними. Например, может быть такая задача: перенести товар их зоны А в зону Б.

Чтобы определить где находится товар, нужно знать его адрес. Для этого есть таблица Addresses. Он включает зону хранения (таблица Areas), номер стелажа (таблица Racks), номер вертикальной секции (таблица VerticalSections) и номер полки (таблица Shelfs).

Товар обычно принадлежит кому-то, как правило это либо физические лица, либо юридические, для них соответственно есть таблицы Individual и LegalEntities.

Таблица Comments содержит комментарии, которые пользователи могут оставлять под задачами.

Для регулярного запуска задачи на проверку просроченных задач склада будет использоваться Hangfire. Hangfire использует несколько таблиц и индексов, для сохранения фоновых заданий и другой информации, связанной с обработкой (представлены на рисунке 4.2).

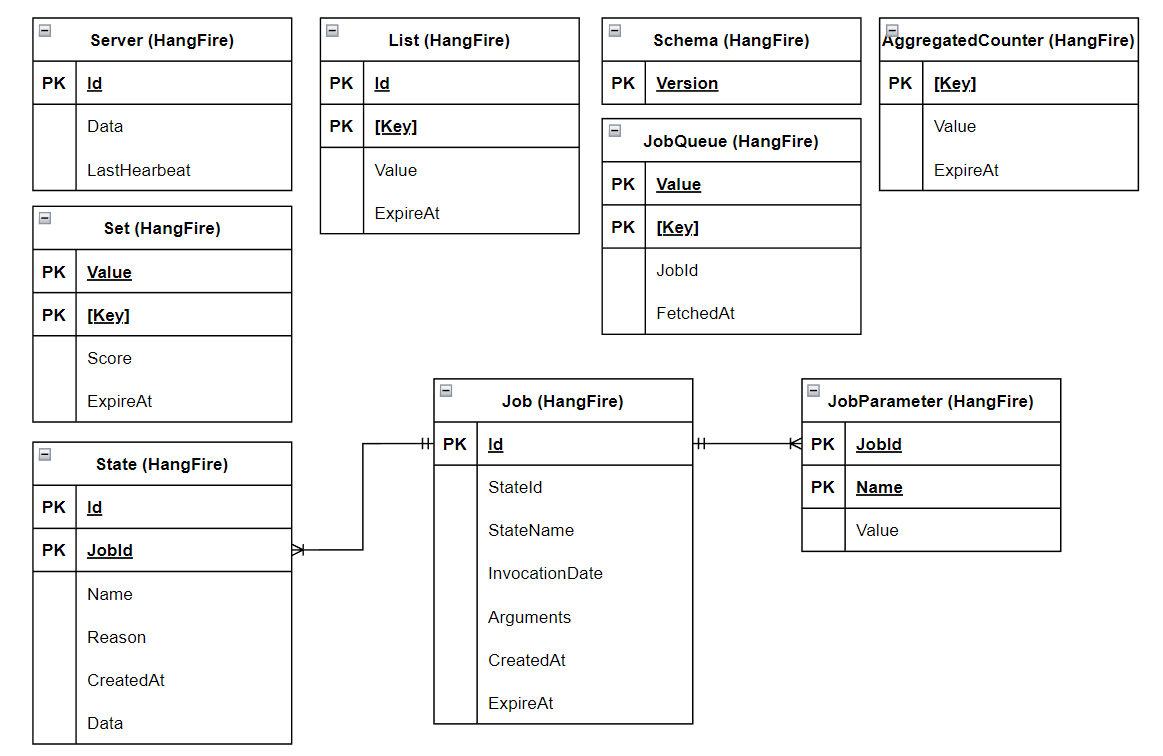


Рисунок 4.2 – Диаграмма базы данных (таблицы Hangfire)

Подробную информацию о таблицах хранилища Hangfire можно посмотреть в официальной документации [4].

Для разработки базы данных был изучен курс лекций Блиновой Е.А. [5].

## **4.2 Архитектура веб-приложения**

Для разрабатываемого веб-приложения была разработана диаграмма развёртывания, представленная на рисунке 4.3.

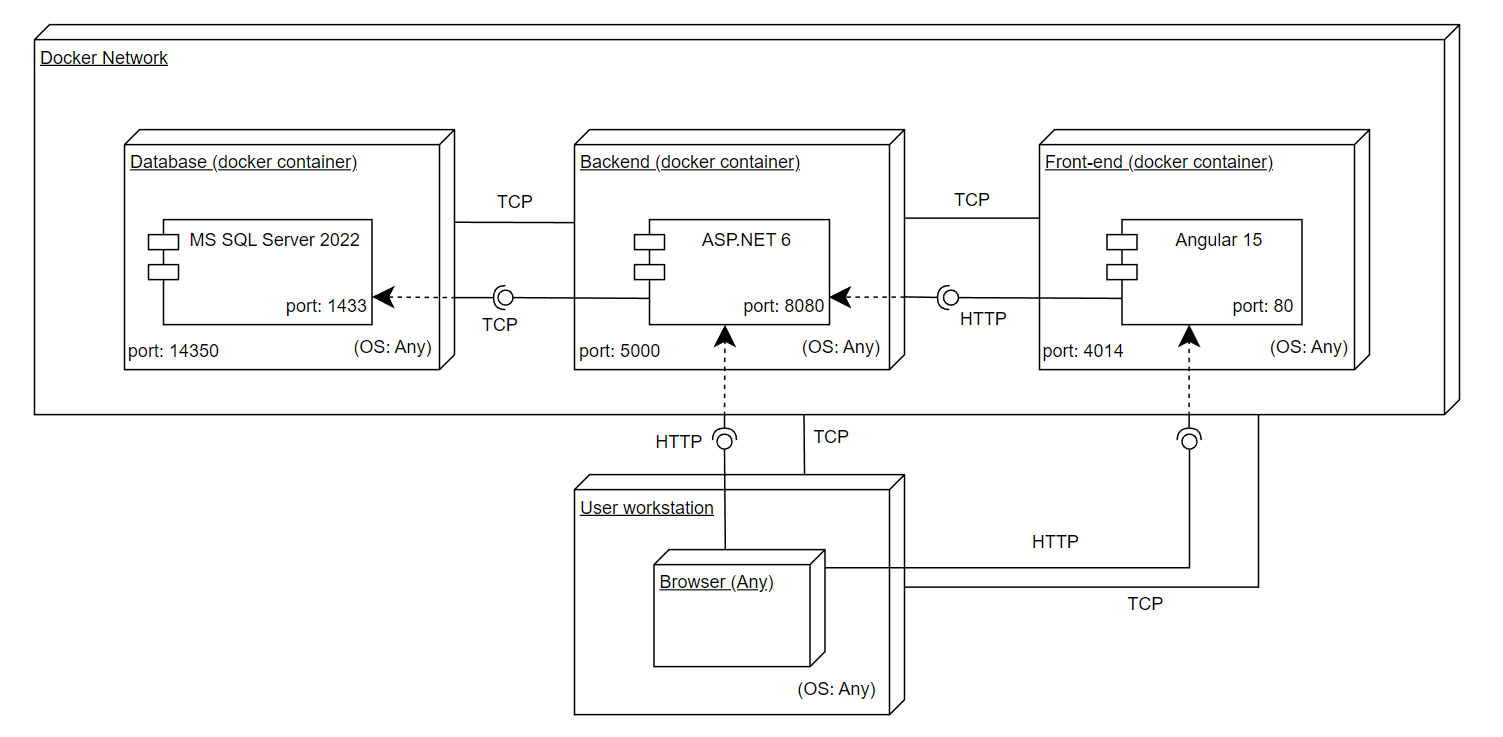


Рисунок 4.3 – Диаграмма развёртывания

Приложение разворачивает в трёх Docker-контейнерах. Контейнер с MS SQL Server 2022, контейнер с ASP.NET 6, контейнер с клиентским приложением на Angular 15.

У каждого контейнера есть внутренние и внешние порты. Backend, Frontend и браузер взаимодействуют по протоколу HTTP, а Backend-приложение и сервер базы данных по TCP.

Для разработки диаграммы развёртывания были использованы материалы курса Парамонова А.И по предмету проектирование информационных систем [6].

## **4.3 Архитектура серверной части веб-приложения**

Архитектура серверной части веб-приложения представлена в виде пакетной диаграммы на рисунке 4.4.

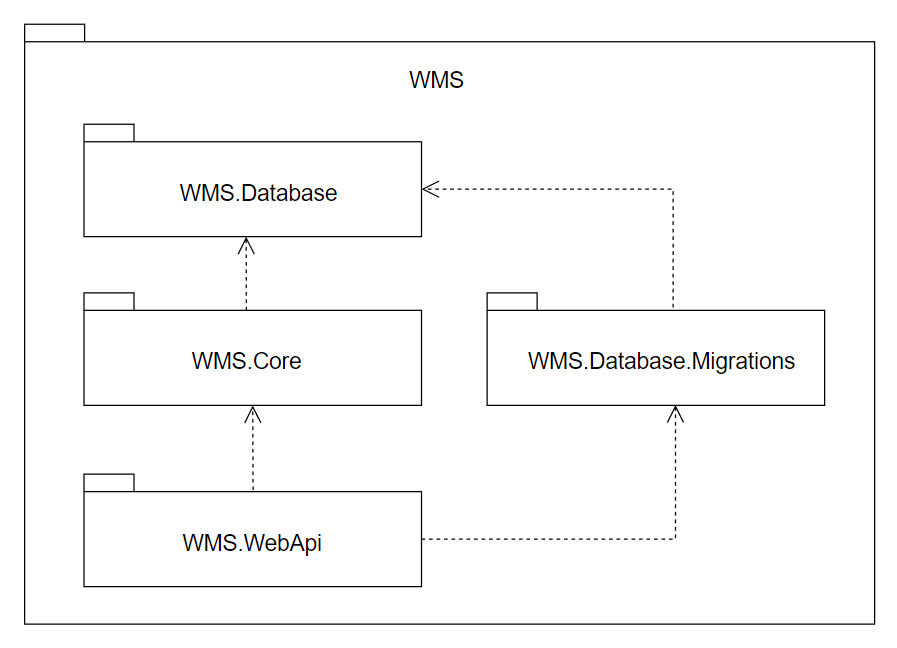


Рисунок 4.4 – Пакетная диаграмма архитектуры серверной части веб-приложения

Архитектура серверной части веб-приложения состоит из четырёх проектов: WMS.Core, WMS.Database, WMS.Database.Migrations и WMS.WebApi.

Проект WMS.WebApi – это ASP.NET приложение типа WebAPI, содержит классы-контроллеры и зависит от проектов WMS.Database, WMS.Database.Migrations, WMS.Core. Из которых импортируются сервисы, сущности и другие вспомогательные классы.

Проект WMS.Migrations – это библиотека классов, содержит классы миграции.

Проект WMS.Database – это также библиотека классов, содержит класс контекста базы данных, классы сущностей, константы для настройки среды и OData, классы конфигурации сущностей.

Проект WMS.Core – содержит сервисы, валидаторы, пользовательские исключения и другие классы, которые реализуют бизнес логику.

## **4.4 Архитектура клиентской части веб-приложения**

Архитектура клиентской части веб-приложения представлена на рисунке 4.5 в виде пакетной диаграммы.

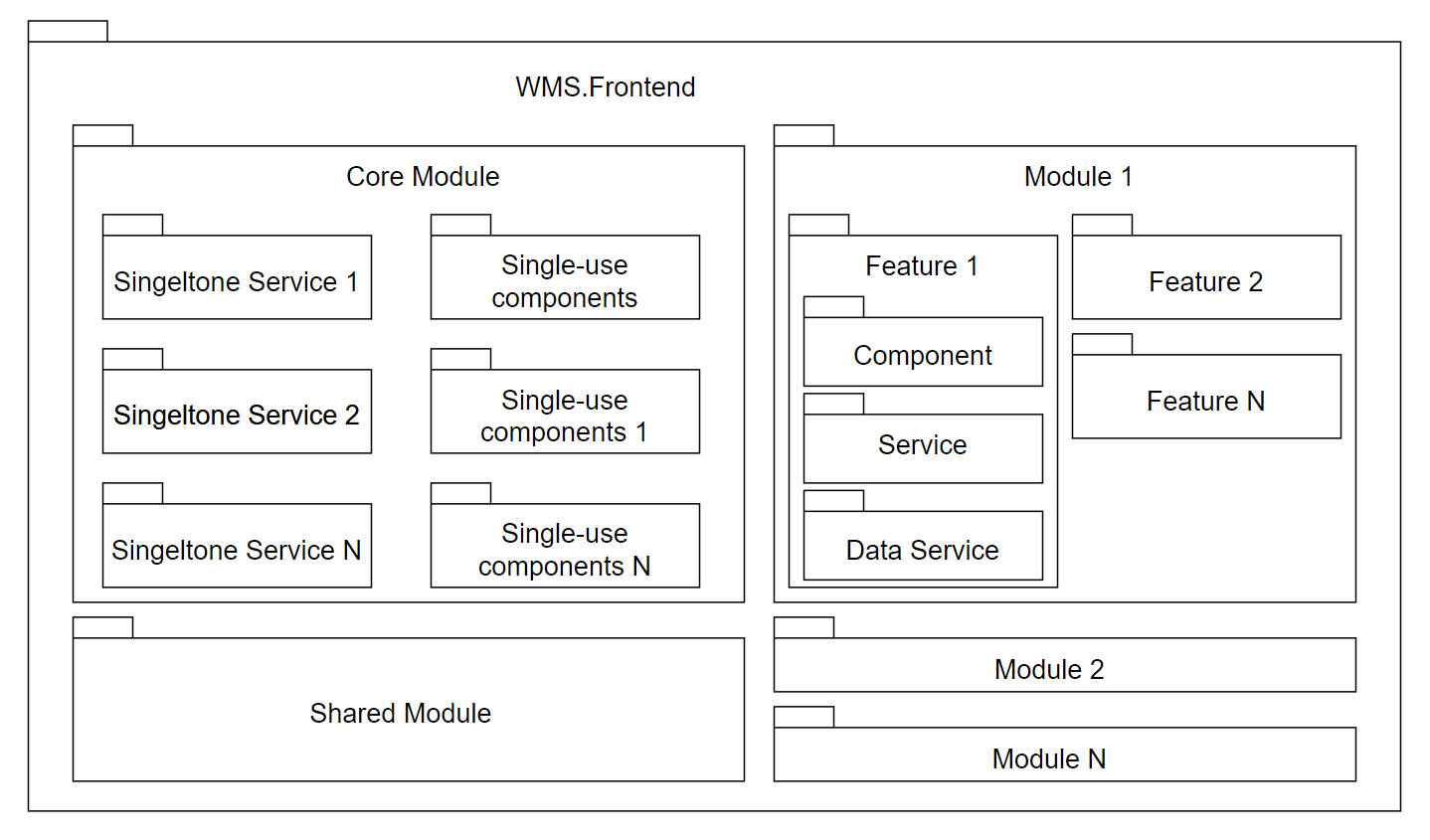


Рисунок 4.5 – Пакетная диаграмма архитектуры клиентской части веб-приложения

Для разработки клиентской части веб-приложения был выбран JavaScript-фреймворк Angular.

Приложение состоит из модулей. Приложение должно иметь корневой модуль. Корневой модуль – это место, где Angular начинает упорядочивать дерево приложения. Модель может импортировать компоненты, которые являются Provider, например, сервисы, хелперы и т.д. Для этого модуль по умолчанию – сингелтонг.

В приложении есть следующие модули и компоненты:

* core, который является основным модулем в приложении, он содержит сингелтон сервисы и одноразовые компоненты, например layout;
* shared, который использует для экспорта компонент, которые делятся между различными модулями (например диалог подтверждение);
* feature, который содержит сервисы, константы, модели и компоненты;
* класс service, который описывает логику компоненты;
* класс data service, который реализует коммуникацию между хранилищем данных и компонентой.

## **4.5 Проектирование пользовательского интерфейса**

Описание компонентов пользовательского интерфейса приведено в таблице 4.1.

Таблица 4.1 – Описание базовых компонент пользовательского интерфейска

|  |  |
| --- | --- |
| Название | Описание |
| login | Форма входа в систему. |
| employees | Список пользователей с режимами чтения, редактирования, создания. |
| individuals | Компонент для управления физическими лицами. |
| legal-entities | Компонент для управления юридическими лицами. |
| areas | Компонент для управления зонами склада. |
| racks | Компонент для управления секциями склада. |
| dictionary-page | Компонент для управления словарями склада. |
| problems | Компонент с доской задач. |
| problem-info | Компонент с подробной информацией о задаче. |
| problem-dialog | Модальное окно с формой для редактирования/создания задачи. |
| problem-comments | Компонент со списком комментариев задачи (должна быть пагинация) и формой для добавления нового комментария. |
| problem-children | Компонент со списком дочерних задач. |
| problem-card | Компонент с краткой информацией о задаче для отображения в виде карточки на доске задач. |
| problem-assign-dialog | Модальное окно с выпадающим списком пользователей для назначения исполнителя задачи. |
| confirm-dialog | Модальное окно для подтверждения какого-либо действия в приложении. |
| address-picker | Пользовательский элемент управления для назначения адреса. |
| wares-form | Форма товара с режимами редактирования, просмотра, создания. |
| wares-navigation | Компонент со навигационным списком товаров. |

## **4.6 Вывод по разделу**

Для проекта была выбрана клиент-серверная архитектура. В качестве сервера базы данных был выбран MS SQL Server. Спроектировано 12 таблиц базы данных. Спроектирован REST API, который будет разработан на основе ASP.NET Web API и будет иметь трёхслойную архитектуру. Клиентское приложение будет разработано на основе JavaScript-фреймворка Angular с модульной архитектурой.

# **Заключение**

В результате прохождения преддипломной практики было разработано веб-приложение, представляющее собой систему управления складом.

В ходе практики был изучен стандарт БГТУ по дипломному проектированию; проведён анализ информационных источников; спроектирована структура базы данных; спроектирована архитектура веб-приложения; разработано REST API; разработан сервис для отправки уведомлений об изменении статуса задачи, регистрации пользователя, приближающейся просрочки выполнения задач; продуман и разработан дизайн пользовательского интерфейса; разработано клиентское приложение; протестировано веб-приложение.

Функционально приложение позволяет поддерживать:

* роли администратора, работника и аудитора;
* общие функции: создание, редактирование, удаление, назначение, комментирование задач; просмотр задача, товаров, словарей;
* функции администратора: создание, редактирование и удаление пользователей; создание, редактирование, удаление товаров;
* подтверждение аудитором готовности задачи;
* уведомление администратора о просрочке задач, изменении статуса задач, нового пользователя о регистрации в системе.

Исходя из полученных результатов можно сделать вывод, что приложение разработано в соответствии с техническим заданием, условия которого выполнены.

# **Список использованных источников**

1 NetSuite [Электронный ресурс]. – Режим доступа: https://www.netsuite.com/portal/home.shtml – Дата доступа 15.04.2023.

2 Sortly [Электронный ресурс]. – Режим доступа: https://www.sortly.com – Дата доступа 15.04.2023.

3 Maropost [Электронный ресурс]. – Режим доступа: https://www.maropost.com – Дата доступа 15.04.2023.

4 Hangfire [Электронный ресурс]. – Режим доступа: https://docs.hangfire.io/en/latest/configuration/using-sql-server.html – Дата доступа 15.04.2023.

6 Блинова Е.А. Курс лекций по базам данных / Е.А. Блинова – Минск: БГТУ, 2022. – 78 с.

7 Парамонов, А.И. Курс лекций по проектированию информационных систем / А.И. Парамонов – Минск: БГТУ, 2022. – 63 с.