

СОДЕРЖАНИЕ

| | |
|---|----|
| ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ | 4 |
| ВВЕДЕНИЕ | 5 |
| 1 Анализ предметной области..... | 8 |
| 2 Постановка задачи..... | 16 |
| 3 Общее описание системы | 18 |
| 3.1 Описание структурной схемы..... | 18 |
| 3.2 Описание функциональной схемы | 20 |
| 4 Описание видов обеспечения..... | 23 |
| 4.1 Математическое обеспечение | 23 |
| 4.2 Информационное обеспечение | 26 |
| 4.3 Лингвистическое обеспечение..... | 29 |
| 4.4 Программное обеспечение | 28 |
| 4.5 Техническое обеспечение..... | 28 |
| 4.6 Методическое обеспечение | 29 |
| 5 Защита информации..... | 30 |
| 6 Практическая реализация | 33 |
| ЗАКЛЮЧЕНИЕ | 37 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 38 |
| ПРИЛОЖЕНИЕ А (обязательное) Структурная схема | 39 |
| ПРИЛОЖЕНИЕ Б (обязательное) Функциональная схема | 40 |
| ПРИЛОЖЕНИЕ В (обязательное) Даталогическая модель | 41 |
| ПРИЛОЖЕНИЕ Г (обязательное) Пример работы программы | 42 |

| | | | | | | | | |
|-----------|-------|----------|-------|------|--|-------------------|-------|--------|
| | | | | | ТГТУ.09.03.01.025 ТЗ-ПЗ | | | |
| Изм. | /лист | № докум. | Подп. | Дата | | | | |
| Разраб. | | Хрестин | | | Разработка системы управления запасом склада с эффективной схемой хранения товаров. Пояснительная записка | Ит. | /лист | Листов |
| Пров. | | Одухов | | | | | 1 | 39 |
| Н. Контр. | | Одухов | | | | САПР, гр. БВТ-191 | | |
| Утв. | | Коробова | | | | | | |

ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

Фреймворк — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

SQL (Structured Query Language) — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

SDK (Software Development Kit) — набор инструментов для разработки программного обеспечения в одном устанавливаемом пакете.

IDE (Integrated Development Environment) — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО). Включает в себя: текстовый редактор, транслятор, средства автоматизации сборки.

СУС — система управления складом.

СУЗ — система управления запасом.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 2 |

ВВЕДЕНИЕ

Современная бизнес-среда характеризуется жесткой конкуренцией и быстро меняющимися требованиями клиентов. В результате предприятия постоянно ищут пути повышения эффективности и прибыльности своей деятельности. Одной из областей, которая может оказать значительное влияние на успех предприятия, является система управления запасами. Эффективное управление запасами может помочь предприятиям минимизировать риск возникновения товарных запасов, сократить расходы на хранение и заказы, а также повысить уровень удовлетворенности клиентов.

Управление запасами – это сложный процесс, требующий тщательного планирования, организации и контроля. В современной розничной торговле малые предприятия сталкиваются с проблемами эффективного управления запасами из-за ограниченности ресурсов и отсутствия сложных систем управления запасами. Поэтому существует потребность в доступных и удобных системах управления запасами, которые могут помочь малому бизнесу повысить эффективность работы и прибыльность. Целью данного диплома является удовлетворение этой потребности путем разработки системы управления запасами (СУЗ) для небольшого магазина с одним складом.

Целью данного диплома является программирование WMS, которая автоматизирует и оптимизирует процесс управления запасами, повышая операционную эффективность и прибыльность магазина. WMS будет разработана с учетом специфических потребностей магазина для обеспечения точного обновления запасов в режиме реального времени. Целью данного диплома является разработка удобной и эффективной WMS, которая может быть легко использована сотрудниками магазина, включая менеджера магазина, продавцов и работников склада.

Предметом данного исследования является процесс управления запасами в небольшом магазине с одним складом. Объектом данного исследования является

| | | | | | | |
|------|------|----------|-------|------|-----------------------------------|------|
| | | | | | <i>ТГТУ.09.03.01.025 КР ТЭ-ПЗ</i> | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 3 |

розничная торговля, в частности, малые розничные предприятия, которые сталкиваются с проблемами управления запасами. Предметом исследования будет разработка СУЗ, которая является доступной, удобной и масштабируемой, что делает ее идеальным решением для малого бизнеса.

Научная новизна данного диплома заключается в разработке СУЗ, специально предназначенной для малого розничного бизнеса. Система будет разработана с использованием языка программирования Python и оптимизирована с помощью математических моделей и алгоритмов. В дипломе также будет рассмотрена постановка и решение проблемы оптимизации, чтобы система обеспечивала точное обновление запасов в режиме реального времени.

Практическая значимость данного диплома заключается в его способности повысить операционную эффективность и прибыльность магазина. СУЗ обеспечит точное обновление запасов в режиме реального времени, сократит количество ошибок и неэффективности, а также повысит удовлетворенность клиентов. Масштабируемость системы означает, что она может быть расширена для удовлетворения будущих потребностей магазина и других малых предприятий. Система будет разработана с целью снижения затрат и времени, связанных с управлением запасами, тем самым повышая прибыльность магазина.

Методы научного исследования, используемые в данном дипломе, включают анализ данных, сбор требований, разработку программного обеспечения и тестирование. Структура диплома будет состоять из нескольких частей, включая анализ домена, постановку задачи, общее описание системы, математическое обеспечение и практическую реализацию. В дипломе также будут представлены руководства пользователя и техническая поддержка для сотрудников магазина.

В заключение следует отметить, что целью данного диплома является решение проблем, с которыми сталкиваются малые розничные предприятия при эффективном управлении товарными запасами. СУЗ, разработанная в данном исследовании, обеспечит доступное, удобное и масштабируемое решение для

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 4 |

повышения операционной эффективности и прибыльности магазина. Структура диплома будет охватывать все аспекты процесса разработки, представляя собой всеобъемлющее руководство для предприятий, стремящихся улучшить свои системы управления запасами. Разработанная в данном исследовании СУЗ будет иметь практическое значение в розничной торговле, способствуя развитию эффективных и экономически выгодных систем управления запасами для малого бизнеса.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 5 |

1 Анализ предметной области

Управление складом играет важную роль в логистической цепочке. Каждое предприятие, занимающееся производством или продажей товаров, сталкивается с необходимостью хранения, учета и управления складом. Управление складом включает в себя различные операции, такие как прием товаров на склад, их хранение, отбор, упаковка и отгрузка. Кроме того, не менее важно контролировать уровень запасов на складе, чтобы вовремя пополнять их, не допуская их недостатка или избытка.

Системы управления складом позволяют автоматизировать и ускорить процессы на складе, обеспечивая точный учет товаров на складе, отслеживание их движения и контроль уровня запасов. Существует множество решений, некоторые из которых предназначены для больших складов с большим объемом товаров, а другие – для небольших складов с небольшим товарооборотом.

Управление запасами является важной частью управления складом. Управление запасами включает в себя определение оптимального уровня запасов, управление заказами и пополнением запасов, а также контроль запасов. Оптимизация управления запасами позволяет сократить издержки, связанные с хранением и управлением запасами, а также сократить риски связанные с избыточными запасами.

Системы управления складом (СУС) – это программное обеспечение, которое позволяет автоматизировать процессы на складе и управлять всеми операциями, связанными с хранением и учетом товаров на складе. СУС позволяет значительно повысить эффективность управления складом, уменьшить ошибки и снизить затраты на управление складом.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 6 |

СУС включает в себя следующие функциональные модули:

- модуль приема товаров на склад – этот модуль позволяет автоматизировать процесс приема товаров на склад, включая проверку товаров на соответствие заказу и их регистрацию в системе управления складом.

- модуль хранения товаров – этот модуль позволяет управлять процессом размещения товаров на складе, включая определение места хранения, контроль запасов и оповещение о необходимости пополнения запасов.

- модуль отбора товаров – этот модуль позволяет автоматизировать процесс отбора товаров с полок для их отгрузки, включая определение оптимального маршрута перемещения товаров и контроль остатков на складе.

- модуль упаковки и отгрузки товаров – этот модуль позволяет управлять процессом упаковки товаров и их отгрузки, включая формирование товаросопроводительной документации и контроль корректности отгрузки товаров.

- модуль управления запасами – этот модуль позволяет автоматизировать процесс управления запасами на складе, включая определение оптимального уровня запасов, управление заказами и контроль запасов.

СУС подразделяются на разные типы, включая:

- локальные системы – это системы, которые установлены непосредственно на компьютерах на складе и работают только в пределах локальной сети.

- облачные системы – это системы, которые доступны через интернет и хранятся на удаленных серверах. Облачные системы позволяют работать с системой управления складом из любой точки мира и не требуют больших затрат на оборудование и его обслуживание.

- гибридные системы – это системы, которые объединяют в себе возможности локальных и облачных систем. Гибридные системы позволяют использовать преимущества обоих типов систем и обеспечивают высокую гибкость в работе с системой управления складом.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 7 |

При выборе СУС необходимо учитывать особенности конкретной компании, ее размеры и объемы товарооборота, а также функциональные требования к системе управления складом.

Преимущества использования СУС включают:

- увеличение эффективности управления складом и сокращение затрат на управление им.
- уменьшение ошибок и повышение точности учета товаров на складе.
- ускорение процессов на складе и повышение производительности труда.
- улучшение качества обслуживания клиентов.
- увеличение прозрачности и контроля над процессами на складе.
- возможность проводить анализы и получать отчеты о работе склада, что позволяет оптимизировать процессы управления складом.

Тем не менее, использование СУС также имеет свои недостатки, такие как:

- высокие затраты на внедрение системы и обучение персонала.
- необходимость постоянного обновления системы и ее адаптации к изменениям в бизнес-процессах компании.
- ограниченность функциональности системы и ее невозможность учитывать все особенности работы компании.
- необходимость постоянного обслуживания и технической поддержки системы.

В целом, использование системы управления складом является одним из важных элементов оптимизации работы склада и всей логистической цепочки компании. Правильный выбор системы и ее адаптация к особенностям компании позволяют снизить затраты на управление складом и повысить его эффективность.

Управление запасами – это процесс управления запасами товаров на складе с целью обеспечения непрерывности производственных и торговых операций компании. Основная задача управления запасами заключается в определении

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 8 |

оптимального уровня запасов, который позволит компании удовлетворять потребности своих клиентов, минимизировать затраты на хранение запасов и избежать потерь, связанных с устареванием товаров.

Основным инструментом управления запасами является система управления запасами (СУЗ), которая позволяет компаниям эффективно управлять запасами, контролировать их движение и максимально оптимизировать процесс управления запасами.

Система управления запасами включает в себя следующие основные компоненты:

- управление заказами — этот компонент включает в себя определение оптимального уровня запасов и управление заказами на закупку товаров. В зависимости от модели управления запасами компания может использовать различные методы определения оптимального уровня запасов, как, например, классификация товаров на складе по степени их значимости для компании.

- управление поставками — этот компонент включает в себя управление процессом поставки товаров на склад, контроль качества поставляемых товаров и управление отношениями с поставщиками.

- управление складом — этот компонент включает в себя управление процессом хранения товаров на складе, контроль запасов и управление процессом отгрузки товаров.

- управление продажами — этот компонент включает в себя управление процессом продажи товаров, контроль запасов и управление процессом отгрузки товаров клиентам.

- управление производством — этот компонент включает в себя управление процессом производства товаров, контроль запасов и управление процессом отгрузки готовой продукции клиентам.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 9 |

- анализ и отчетность — этот компонент включает в себя анализ производительности системы управления запасами и получение отчетов о работе системы.

Система управления запасами позволяет компаниям оптимизировать процесс управления запасами и достичь следующих преимуществ:

- снижение затрат на хранение запасов и уменьшение потерь, связанных с устареванием товаров.

- улучшение уровня обслуживания клиентов и сокращение времени поставки товаров.

- увеличение эффективности производственных и торговых операций компании.

- увеличение гибкости и масштабируемости производства и торговых операций.

- улучшение контроля над процессом управления запасами и повышение прозрачности в работе всей логистической цепочки компании.

- возможность проводить анализы и получать отчеты о работе системы управления запасами, что позволяет оптимизировать работу всей логистической цепочки компании.

Однако использование системы управления запасами также имеет свои недостатки, такие как:

- высокие затраты на внедрение системы и обучение персонала.

- необходимость постоянного обновления системы и ее адаптации к изменениям в бизнес-процессах компании.

- ограниченность функциональности системы и ее невозможность учитывать все особенности работы компании.

- необходимость постоянного обслуживания и технической поддержки системы.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 10 |

Размещение товаров на складе – это один из важнейших этапов логистической деятельности компании, который напрямую влияет на эффективность процесса управления запасами и уровень обслуживания клиентов.

Основные принципы размещения товаров на складе:

- размещение товаров должно быть организовано таким образом, чтобы обеспечить максимальную эффективность использования площади склада и минимизировать время на поиск и отбор товаров.
- товары должны быть размещены на складе с учетом их характеристик и особенностей хранения, а также с учетом частоты и объема продаж.
- на складе должна быть организована система маркировки и идентификации товаров, которая позволит быстро находить и отбирать нужные товары.
- размещение товаров на складе должно соответствовать требованиям пожарной безопасности и санитарно-гигиеническим нормам.

Существует несколько методов размещения товаров на складе:

- Метод кросс-докинга – это метод, при котором товары не размещаются на складе, а сразу после прибытия на склад перегружаются на транспортное средство для доставки клиенту. Этот метод позволяет сократить время на перевалку товаров и повысить скорость доставки товаров.
- Метод зонирования – это метод, при котором на складе создаются зоны с различными условиями хранения для разных групп товаров. Например, зона с низкой температурой для хранения продуктов питания, зона с малой влажностью для хранения электроники и т.д.
- Метод сезонного размещения – это метод, при котором товары размещаются на складе в зависимости от сезона и спроса на них. Например, зимние товары размещаются на складе летом, а летние – зимой.
- Метод размещения товаров по частоте продаж – это метод, при котором товары размещаются на складе в порядке убывания частоты продаж. Товары,

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 11 |

которые чаще всего продается, размещаются ближе к зоне отбора, чтобы сократить время на их отбор.

Правильно организованное размещение товаров на складе позволяет компаниям снизить затраты на управление запасами, сократить время на обработку заказов и повысить уровень обслуживания клиентов.

Отбор товаров – это процесс, при котором сотрудники склада собирают заказы клиентов из имеющихся на складе товаров. Этот процесс является одним из ключевых элементов логистической цепочки компании, так как от его эффективности зависит уровень обслуживания клиентов и время доставки товаров.

Основные принципы отбора товаров:

- отбор товаров должен быть организован таким образом, чтобы минимизировать время на отбор заказов и повысить скорость доставки товаров клиентам.

- сотрудники склада, занимающиеся отбором товаров, должны иметь высокий уровень компетенции и знаний о характеристиках и особенностях хранения товаров.

- на складе должны быть организованы зоны отбора товаров, которые позволят сотрудникам склада быстро находить нужные товары.

- отбираемые товары должны быть тщательно проверены на соответствие заказу и наличие дефектов.

Существует несколько методов отбора товаров:

- Отбор товаров по заказу – это метод, при котором сотрудники склада отбирают товары для каждого заказа отдельно. Этот метод позволяет минимизировать ошибки при отборе товаров и повысить точность выполнения заказов.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 12 |

- Отбор товаров по партиям – это метод, при котором товары отбираются пакетами из нескольких заказов. Этот метод позволяет сократить время на отбор товаров и уменьшить количество ошибок при отборе.

- Отбор товаров по зонам – это метод, при котором товары отбираются из зон, организованных на складе. Этот метод позволяет сократить время на поиск товаров и повысить скорость отбора товаров.

В процессе отбора товаров в основном используются два технологических решения:

- Системы сканирования штрих-кодов – это система, которая позволяет сотрудникам сканировать штрих-коды на товарах для идентификации товара и сверки с заказом.

- Автоматические системы отбора товаров – это система, которая позволяет автоматически отбирать товары из зон на складе без участия сотрудников склада.

Правильно организованный процесс отбора товаров позволяет компаниям повысить эффективность работы склада, снизить время на обработку заказов и улучшить уровень обслуживания клиентов.

Управление складом является важным элементом логистической цепочки компаний, который позволяет при правильной организации эффективно управлять запасами товаров, минимизировать затраты на хранение и улучшать уровень обслуживания клиентов. Все вышеописанные принципы должны быть адаптированы к конкретным потребностям и характеристикам каждой компании. Таким образом можно достичь максимальной эффективности управления складом и повысить конкурентоспособность компании на рынке.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 13 |

2 Постановка задачи

Целью данной работы является повышение эффективности комплектации и маршрутизации продукции на складе. Для достижения поставленной цели необходимо разработать систему управления запасами на складе. В том числе должны быть решены следующие задачи:

- разработка схемы базы данных и создание необходимых таблиц для хранения информации о товарных позициях и их местонахождении на складе, местах хранения, носителях товаров и текущем уровне запасов;

- разработка пользовательского интерфейса, который обеспечит удобный интерфейс для управления инвентарными позициями и их местоположением, местами хранения, носителями товаров и уровнем запасов;

- реализация функциональности CRUD (Create, Read, Update, Delete) для управления сущностями созданных таблиц;

- разработка функциональности для регистрации инвентарных операций, таких как получение новых инвентарных позиций, размещение их на складе и отбор для продажи;

- Внедрение функциональности отслеживания запасов, которая позволит системе отслеживать уровни запасов в режиме реального времени и предоставлять отчеты об уровнях запасов и их движении.

В процессе анализа существующих решений был определен функционал, который должен быть реализован в разрабатываемой системе.

На следующем этапе осуществляется выбор целевой платформы и определение необходимых вычислительных ресурсов для работы системы. В качестве платформы для данной системы будем использовать веб-приложение, т.к. браузерные приложения гибкие, универсальные, не требуют предварительной подготовки среды, позволяют сэкономить финансы компании, аппаратные ресурсы, время сотрудников.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 14 |

После выбора платформы необходимо выбрать стек технологий (набор программных решений и библиотек, которые будут обеспечивать функционирование сервиса), который будет использоваться для разработки каждой части системы. Для этого необходимо провести анализ и выявить, какой язык программирования лучше подходит для решения поставленной задачи и какие возможности он предоставляет.

Информационная и даталогическая модели, функциональная схема разрабатываемого решения строятся, исходя из заложенного в систему функционала. Информационная модель отражает общую структуру сущностей, присутствующих в системе, не вдаваясь в подробности реализации каждой из них. Даталогическая модель раскрывает внутреннюю структуру сущностей, а также устанавливает связи между ними. Функциональная схема описывает все типовые сценарии взаимодействия пользователей с системой.

Основываясь на функциональной схеме и даталогической модели, наступает этап практической реализации. Разработка веб-приложения требует развертывания серверной части для тестирования разрабатываемого ПО, поэтому предварительно необходимо настроить необходимое ПО. После этого в соответствии с даталогической моделью создается база данных. Далее разрабатываются все модули системы. После окончания процесса разработки производится тестирование работоспособности системы. При возникновении ошибок необходимо произвести анализ и устранить их. На последнем этапе создается справочная документация по работе приложения.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 15 |

3 Общее описание системы

Архитектура системы является клиент-серверной, где клиентом является интерфейс пользователя для управления запасами, поставками и комплектацией, а сервером – база данных и модули управления запасами, поставками и комплектацией.

Архитектура системы управления складом представляет собой трехуровневую структуру. Это позволит разделить функциональность системы на три уровня: уровень представления, уровень бизнес-логики и уровень доступа к данным.

3.1 Описание структурной схемы

В структурной схеме представлены подсистемы, их информационные обеспечения, лингвистические обеспечения, программные и математическое обеспечение, а также математическое обеспечение для всей программы и техническое обеспечение для ее корректной работы. Структурная схема представлена в приложении А.

В систему входят следующие подсистемы:

- Подсистема представления – на уровне представления будет реализован интерфейс пользователя, который позволит управлять складскими процессами и контролировать запасы товаров. На этом уровне будет реализована функциональность управления запасами, учета товаров, контроля за сроками годности и др. Для реализации интерфейса пользователя будем использовать технологии веб-разработки, а именно HTML, CSS с использованием шаблонов Django и фреймворка Bootstrap.

- Подсистема бизнес-логики – на уровне бизнес-логики будет реализована основная функциональность системы управления складом. На этом уровне будут реализованы модули управления запасами, учета товаров, контроля за сроками

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 16 |

годности и др. Для реализации бизнес-логики системы будем использовать язык программирования Python.

- Подсистема доступа к данным – На уровне доступа к данным будет реализована функциональность доступа к базе данных. Для хранения данных будем использовать реляционную базу данных PostgreSQL. Для доступа к базе данных будем использовать язык программирования Python.

Для правильной работы техническое обеспечение для данной системы должно составлять для каждого из уровней.

- Уровень представления:
- Персональный компьютер со следующими характеристиками:
 - Процессор Intel Core i3 или выше;
 - ОЗУ от 4 ГБ;
 - Видеокарта с поддержкой OpenGL 2.0 или выше;
 - Клавиатура и мышь.
- Терминал сбора данных (ТСД) CipherLab RK95-2S-38K.
- Уровень бизнес-логики:
 - Персональный компьютер со следующими характеристиками:
 - Процессор Intel Core i7 или выше;
 - ОЗУ от 16 ГБ;
 - Жесткий диск SSD;
 - Графический адаптер с поддержкой OpenGL 3.3 или выше.
- Уровень доступа к данным:
 - Персональный компьютер со следующими характеристиками:
 - Процессор Intel Core i5 или выше;
 - ОЗУ от 8 ГБ;
 - Жесткий диск SSD;

- Уровень разработки системы:

- Среда разработки IntelliJ IDEA.

Лингвистическое обеспечение системы:

- язык программирования Python;

- фреймворк Django;

- язык гипертекстовой разметки HTML;

- формальный язык описания веб-страницы CSS;

- фреймворк Bootstrap;

- SQL запросы;

- пользовательский интерфейс – элементы навигации, диалоги, меню, формы.

Информационное обеспечение системы:

- таблица «storage_unit»;

- таблица «products»;

- таблица «inventory_item»;

- таблица «invernotry_object»;

- таблица «inventory_transaction».

- таблица «employees»;

3.2 Описание функциональной схемы

Функциональная схема представлена в приложении Б. Функциональную схему разрабатываемой СУЗ можно разделить на три основных компонента: управление запасами, отчетность и пользовательский интерфейс. Рассмотрим каждый компонент подробнее.

1. Управление запасами.

Компонент управления запасами является ядром системы. Он позволяет пользователям управлять запасами, включая получение, складирование и комплектацию продукции. Компонент управления запасами состоит из следующих подкомпонентов:

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 18 |

- Получение продуктов: этот подкомпонент позволяет пользователям получать продукты на склад. Пользователи могут вводить в систему данные о полученных продуктах, такие как название продукта, штрих-код, количество и местоположение. Система обновляет уровни запасов и добавляет записи транзакций для отслеживания движения запасов;

- Запасы продуктов: этот подкомпонент позволяет пользователям размещать товары на паллетах. Пользователи могут выбрать продукт, который они хотят складировать, ввести местоположение паллета и количество продукта, который они складывают. Система обновляет уровни запасов и добавляет записи операций для отслеживания движения запасов;

- Подбор продуктов: этот подкомпонент позволяет пользователям выбирать продукты с полок. Пользователи могут выбрать продукт, который они хотят выбрать, ввести расположение паллета и количество продукта, который они выбирают. Система обновляет уровни запасов и добавляет записи транзакций для отслеживания движения запасов.

2. Отчетность.

Компонент отчетности СУЗ предоставляет отчеты в режиме реального времени об уровне запасов, движении запасов и производительности продукции. Компонент отчетности предоставляет отчеты в режиме реального времени об уровне запасов, движении запасов и производительности продукции. Пользователи могут создавать отчеты на основе различных критериев, таких как название продукта, штрих-код, местоположение и дата.

3. Пользовательский интерфейс.

Компонент СУЗ с веб-интерфейсом пользователя позволяет сотрудникам управлять запасами, включая получение, складирование и отбор продукции. Он состоит из следующих страниц:

- список инвентарных позиций: отображает список всех инвентарных позиций в системе. Пользователи могут просматривать подробную информацию

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | 19 |
| Изм. | Лист | № докум. | Подп. | Дата | | |

о каждой позиции, включая ее название, штрих-код, местоположение и количество;

- создание инвентарной позиции: позволяет пользователям создавать новые инвентарные позиции. Пользователи могут ввести сведения о новом элементе, включая его название, штрих-код, местоположение и количество;

- детали инвентарного объекта: отображает подробную информацию об инвентарном объекте, включая его название, штрих-код, местоположение, количество, минимальное количество, максимальное количество и историю транзакций;

- обновление инвентарной позиции: позволяет пользователям обновлять информацию о существующей инвентарной позиции, включая ее название, штрих-код, местоположение, количество, минимальное количество и максимальное количество;

- удалить инвентарную позицию: позволяет пользователям удалять существующие инвентарные позиции из системы;

- создание инвентарной операции: позволяет пользователям добавлять операции для инвентарной позиции, включая тип операции (ввоз/вывоз) и количество.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | 20 |
| Изм. | Лист | № докум. | Подп. | Дата | | |

4 Описание видов обеспечения

4.1 Математическое обеспечение

Для оптимизации маршрутизации продукции на складах необходимо решить задачу кратчайшего пути. Проблема кратчайшего пути представляет собой задачу поиска оптимального пути между двумя узлами в графе с минимальной суммой весов ребер. В контексте управления запасами, для оптимизации процесса маршрутизации, необходимо найти оптимальный маршрут между множеством узлов на складе, а не только между двумя. Эта задача может быть решена с помощью алгоритма Дейкстры.

Алгоритм Дейкстры – это алгоритм поиска кратчайшего пути от одной вершины графа до всех остальных вершин. Он работает с положительными весами ребер и основан на поочередном добавлении вершин в множество S , которые имеют минимальное расстояние от исходной вершины. Алгоритм может быть использован задачи оптимизации, так как граф маршрутизации на складе является взвешенным и неориентированным. Сложность этого алгоритма составляет

$$\theta(E + V \cdot \log(V)), \quad (1)$$

где: E – количество ребер, V – количество вершин.

4.1.1 Математическая модель

Математическая модель для решения задачи маршрутизации продукции на складе описывается графом, в котором вершины соответствуют областям на складе, а ребра графа определяют длины путей между вершинами. Для алгоритма Дейкстры математическая модель представлена следующим образом:

- V – множество вершин графа, где каждая вершина представляет собой определенную область на складе.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 21 |

- E – множество ребер графа, где каждое ребро представляет длину пути между вершинами.

- $w(u,v)$ – функция веса ребра графа, которая определяет расстояние между вершинами u и v .

- s – начальная вершина графа, которая представляет собой начальное положение продукции на складе.

- t – конечная вершина графа, которая представляет собой конечное положение продукции на складе.

4.1.2 Постановка математической задачи

Формулировка задачи маршрутизации продукции на складе может выглядеть следующим образом:

Дано n вершин, где каждая вершина представляет собой определенную область на складе. Каждая вершина имеет свои координаты x_i и y_i . Для каждой пары вершин i и j задана длина пути $dist(i, j)$, которая определяет расстояние между вершинами. Также задано начальное положение продукции на складе и конечное положение, куда продукцию необходимо переместить. Необходимо найти кратчайший путь между начальным и конечным положением продукции на складе, используя заданные длины путей между вершинами.

4.1.3 Алгоритм решения задачи

1. Инициализация:

- устанавливаем расстояние от начальной вершины s до всех остальных вершин графа равным бесконечности:

$$\forall u \in V, u \neq s: dist(s, u) = \infty, \quad (2)$$

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 22 |

- устанавливаем расстояние от начальной вершины s до самой себя равным нулю:

$$dist(s, s) = 0, \quad (3)$$

2. Основной цикл:

- для каждой вершины v , кроме начальной вершины s , находим кратчайший путь от начальной вершины s до v :

$$dist(s, v) = \min\{dist(s, u) + w(u, v)\}, \quad (4)$$

где: u – предыдущая вершина на пути от s до v ;

$w(u, v)$ – вес ребра между вершинами u и v .

- после нахождения кратчайшего пути до вершины v , помечаем ее как обработанную.

3. Алгоритм завершается, когда все вершины графа будут обработаны или когда будет найден кратчайший путь до конечной вершины.

Таким образом, алгоритм Дейкстры для решения задачи оптимизации маршрутизации продукции на складе заключается в нахождении кратчайших путей от начальной вершины до всех остальных вершин графа

4.2 Информационное обеспечение

В процессе проектирования базы данных была разработана ее даталогическая модель. Даталогическая модель разрабатываемой системы представлена в приложении Б.

Данные, необходимые для функционирования системы, хранятся в базе данных «warehouse_db», которая содержит следующие таблицы:

- таблица «storage_units» содержит в себе информацию о местах хранения на складе, их штрих-кодах. Места хранения привязываются к карте склада их идентификационными номерами. Разрабатываемая система имеет два виртуальных места хранения, не имеющих штрих-коды, для упрощения работы с таблицами базы данных: места хранения с идентификационными номерами 0 (если на это место хранения ссылается паллет, это означает, что паллет находится вне склада) и 1 (паллет перемещается по складу в данный момент);

- таблица «products» содержит в себе такую информацию о реализуемой продукции, как штрих-код производителя, название продукта, стоимость, категория продукта, производитель;

- таблица «inventory_objects» содержит в себе информацию о паллетах, их штрих-кодах, местоположения на складе согласно местам хранения, заполненность паллета. Каждый паллет связан с местом хранения через его идентификационный номер;

- таблица «inventory_items» содержит в себе товарные позиции на складе. Товарная позиция описывается штрих-кодом, который привязан к таблице «products», количеством единиц продукта, датой добавления и датой истечения срока действия;

- таблица «inventory_transactions» описывает операции с товарными позициями, такими как разгрузка, приемка, перемещение товаров, и дату

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 24 |

проведения операции. Связка товарных позиций происходит через идентификационный номер из таблицы «inventory_items»;

- таблица «employees» содержит в себе сведения о сотрудниках, работающих на складе. В таблице хранятся имена сотрудников, должности, уровни доступа к системе и учетные данные для входа в систему.

4.3 Лингвистическое обеспечение

Основным языком разработки веб-системы был выбран язык программирования Python. Python это легкий в изучении, мощный язык программирования. Он имеет эффективные высокоуровневые структуры данных и простой, но эффективный подход к объектно-ориентированному программированию.

Django Framework – это один из самых популярных фреймворков для создания веб-приложений на Python. Django является высокоуровневым веб-фреймворком, разработанным на языке программирования Python. Данный фреймворк предназначен для ускорения процесса веб-разработки и обеспечивает чистый и прагматичный дизайн. Он позволяет значительно снизить трудозатраты, связанные с веб-разработкой. Таким образом, Django позволяет разработчикам сосредоточиться на написании собственных приложений, не тратя время на изобретение уже существующих решений.

PostgreSQL – это мощная объектно-реляционная система баз данных с открытым исходным кодом, которая использует и расширяет язык SQL в сочетании с множеством функций, позволяющих надежно хранить и масштабировать самые сложные рабочие нагрузки данных

При разработке клиентской части системы использовались язык разметки HTML и язык описания стилей CSS для создания визуальной части.

Взаимодействие пользователя с серверной частью системы реализуется с использованием графического интерфейса, который состоит из нескольких

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 25 |

страниц. Веб-страницы могут включать в себя элементы, необходимые для просмотра данных (таблицы), а также элементы, необходимые для их редактирования, такие как формы ввода или функциональные кнопки.

4.4 Программное обеспечение

Для разработки веб-приложения на Java необходима среда разработки IntelliJ IDEA by JetBrains. Была выбрана версия Ultimate, которая позволяет использовать возможности Django.

Также для тестирования, отладки и исполнения веб-приложений на основе Python в связке с СУБД PostgreSQL использовался комплект серверных программ Apache HTTP Server.

Для взаимодействия с клиентской частью системы необходимо наличие веб-браузера с поддержкой последних стандартов веб-технологий, таких как HTML5, CSS3, JavaScript.

4.5 Техническое обеспечение

Для функционирования серверной части системы необходим виртуальный или выделенный сервер. Целесообразно использовать виртуальный сервер, так как серверная часть приложения является легковесной, а также использование виртуального сервера дает разработчику полный контроль над развернутым ПО.

Для разработки, отладки и использования необходимо наличие персонального компьютера, клавиатуры, мыши, монитора или ноутбука. Минимальные требования: четырехъядерный 64-разрядный процессор с тактовой частотой 1,6 ГГц или выше, 6 Гб ОЗУ, 40Гб свободного пространства на накопителе.

Для работы с клиентской частью приложения необходим доступ к сети Интернет. Также рекомендовано, но не обязательно наличие терминала сбора

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 26 |

данных. Минимальные требования: процессор с тактовой частотой 1,4 ГГц или выше, 2 Гб ОЗУ, 16 Гб свободного пространства на накопителе, физическая клавиатура 25 или 28 кнопок, Wi-Fi IEEE 802.11 модуль.

4.6 Методическое обеспечение

При первом запуске главной страницы пользователю необходимо пройти аутентификацию, введя в форму логин и пароль. После этого откроется главная страница, в котором расположено меню. Пользователю необходимо выбрать необходимую функциональность, нажав соответствующую кнопку:

- «список позиций»;
- «создание позиции»;
- «редактирование позиции»;
- «удаление позиции»;
- «детали паллета»;
- «создание операции»;
- «детали операций»;
- «карта склада».

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | 27 |
| Изм. | Лист | № докум. | Подп. | Дата | | |

5 Защита информации

В современном мире информация играет важную роль, поэтому защита от утечек конфиденциальной информации является важным аспектом в различных секторах деятельности. Защита информации представляет собой комплекс мер, направленных на предотвращение утечки защищаемых данных и непреднамеренных или несанкционированных воздействий на защищаемые данные. В контексте разработки системы, применены следующие методы защиты информации:

- аутентификация пользователей, с использованием пароля;
- использование протокола HTTPS;
- хэширование паролей пользователей;
- встроенные механизмы защиты фреймворка Django.

Чтобы войти в систему, пользователю необходимо ввести логин и пароль. Если пользователь не зарегистрирован в системе, то администратор имеет возможность зарегистрировать пользователя в системе и предоставить ему право доступа к ее ресурсам, право на изменение информации, хранящейся в базе данных, право на запись новой информации. Данный функционал обеспечивает конфиденциальность информации.

Протокол HTTPS является защищенной версией протокола HTTP, который используется для передачи данных в Интернете. Он обеспечивает безопасную передачу данных между клиентом и сервером, используя шифрование данных и аутентификацию. HTTPS использует криптографические протоколы для шифрования данных, передаваемых между клиентом и сервером, что позволяет обеспечить конфиденциальность информации, передаваемой между клиентом и сервером. HTTPS обеспечивает аутентификацию сервера и клиента, что позволяет убедиться в том, что данные не передаются злоумышленникам, а также обеспечивает защиту от некоторых видов атак, в частности, от атаки типа "человек посередине" (man-in-the-middle).

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 28 |

Хэширование паролей пользователей (процесс преобразования пароля в строку случайных символов, которая служит в качестве проверочного значения вместо самого пароля) позволяет сохранить пароли пользователей в безопасном виде и предотвратить их утечку в случае взлома базы данных или другого способа несанкционированного доступа к информации.

При использовании хэширования паролей, пользователь при регистрации вводит свой пароль, который затем хэшируется и сохраняется на сервере. При входе в систему, введенный пользователем пароль также хэшируется, и полученное значение сравнивается с сохраненным значением хэша. Если значения совпадают, то пользователь получает доступ к системе. Такая процедура позволяет предотвратить утечку паролей в случае несанкционированного доступа к базе данных. Даже если злоумышленники получают доступ к хэшам паролей, они не смогут прочитать их и использовать для входа в систему. На сервере Linux широко используется алгоритм хэширования паролей SHA-512, который обеспечивает высокую степень защиты при правильной настройке и использовании.

Фреймворк Django имеет встроенные механизмы для защиты от атак, таких как SQL-инъекции и XSS-атаки. SQL-инъекция (SQL Injection) - это атака на веб-приложение, которая позволяет злоумышленнику получить несанкционированный доступ к базе данных приложения. Атака происходит путем внедрения вредоносного SQL-кода в форму или URL-адрес, которые обрабатываются приложением и передаются в базу данных.

XSS (Cross-Site Scripting) - это атака на веб-приложение, которая позволяет злоумышленнику внедрять вредоносный код на страницу и заставлять ее выполняться в браузере пользователя. Атака происходит путем внедрения вредоносного скрипта в URL-адрес, форму или любое другое веб-приложение, которое позволяет пользователю вводить данные.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 29 |

Как правило, обе атаки используются для кражи личных данных пользователей, например, логинов и паролей. Злоумышленник может изменить или удалить данные из базы данных, выполнить другие вредоносные действия, такие как перенаправление пользователя на другой сайт или внедрение вредоносного кода на страницу. Django использует специальные методы защиты, такие как фильтрация ввода данных, проверка на валидность вводимых данных, использование параметризованных запросов к базе данных, правильная настройка конфигурации базы данных и сервера, чтобы минимизировать возможность атак.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | 30 |
| Изм. | Лист | № докум. | Подп. | Дата | | |

6 Практическая реализация

6.1 Подготовка среды выполнения

Чтобы начать программировать на Python, нужно установить интерпретатор Python. Интерпретатор выступает в роли дешифровщика исходного кода в байт-код, который впоследствии выполняется. Существует несколько реализаций Python, и среди них для разработки была выбрана реализация CPython, так как она считается эталонной, поддерживает большинство активно используемых платформ и распространяется под свободной лицензией. Была скачана и установлена версия Python 3.10.5.

Разрабатываемая система является веб-приложением. Для разработки серверной части приложения был выбран фреймворк Django. Django - это веб-фреймворк, написанный на языке программирования Python, который предоставляет множество преимуществ. Он позволяет быстро создавать веб-приложения благодаря готовым шаблонам, аутентификации и административному интерфейсу. Фреймворк имеет встроенный ORM, что упрощает работу с базой данных и избавляет от необходимости писать сложные SQL-запросы. Django имеет встроенные механизмы для защиты от атак, таких как SQL-инъекции и XSS-атаки. Фреймворк позволяет создавать крупные и сложные веб-приложения, которые могут обрабатывать большое количество запросов и пользователей.

Для разработки приложения в целом необходимо установить среду разработки. Была выбрана одна из самых популярных IDE на настоящий момент для языка Python – IntelliJ IDEA версии Ultimate. Версия Ultimate была выбрана по следующим соображениям:

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 31 |

- поддержка Django: IntelliJ Idea предоставляет полноценную поддержку Django, включая автоматическое создание и обновление шаблонов, автодополнение для моделей и представлений.

- интеграция с Git: IntelliJ Idea имеет встроенную поддержку Git, позволяющую легко контролировать версии кода, создавать ветки и выполнять слияния.

- удобная отладка: IntelliJ Idea предоставляет мощный отладчик Python, который позволяет удобно отслеживать выполнение кода и исправлять ошибки.

- Возможность развертывания приложения на удаленном сервере и интеграцию с системами управления базами данных.

6.2 Разработка базы данных

Разработка проекта в первую очередь начинается с разработки базы данных, соответствующей требованиям задания. Была выбрана СУБД PostgreSQL. Для её внедрения необходимо установить дистрибутив PostgreSQL на сервер. После установки сервер PostgreSQL будет запущен и будет работать по порту 5432. В дистрибутив входит консольный клиент psql, который позволяет интерактивно вводить запросы, передавать их в PostgreSQL и видеть результаты. С помощью psql создается база данных «inventory_db» и после создания база данных не содержит в себе таблиц. Далее создаются таблицы, которые в дальнейшем будут использоваться в проекте: «storage_unit», «products», «inventory_item», «inventory_object», «inventory_transaction», «employees». Процесс создания базы данных через командную строку показан на рисунке .

6.3 Разработка серверной части

Приложение на Django работает по принципу клиент-серверной архитектуры. Клиент (обычно браузер) отправляет HTTP-запрос на сервер, где

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| Изм. | Лист | № докум. | Подп. | Дата | | 32 |

работает приложение Django. Приложение обрабатывает запрос, взаимодействуя с базой данных и другими компонентами, и возвращает ответ клиенту в виде HTML-страницы. Функционирование приложения Django основано на использовании паттерна проектирования Model-View-Controller (MVC), который позволяет разделить приложение на три логические части:

- Модели (Models) - это слой, отвечающий за работу с базой данных. Здесь определяются объекты и их связи в базе данных.

- Представления (Views) - это слой, отвечающий за обработку запросов и взаимодействие с моделями и шаблонами. Здесь определяется логика обработки запросов и формирования ответов.

- Шаблоны (Templates) - это слой, отвечающий за внешний вид страниц приложения. Здесь определяется внешний вид страниц, который будет отображаться в браузере.

При обработке HTTP-запроса в Django происходит следующая последовательность событий:

1. Сервер принимает HTTP-запрос.
2. Django маршрутизирует запрос на соответствующее представление.
3. Представление вызывает необходимые модели и шаблоны.
4. Модели взаимодействуют с базой данных.
5. Шаблоны создают HTML-страницу, которая возвращается в браузер пользователю.
6. Если в запросе были переданы данные, то они обрабатываются на стороне сервера.
7. Если в запросе было передано что-то неправильное или произошла ошибка, то сервер возвращает соответствующий HTTP-статус и сообщение об ошибке.
8. Если запрос был успешно обработан, то сервер возвращает HTML-страницу с результатами.

9. Браузер отображает полученную HTML-страницу пользователю.

Разработка проекта на Django начинается с разворачивания проекта на сервере, который имеет следующую файловую структуру:

- manage.py: исполняемый файл, который позволяет управлять проектом.
- Название проекта/: директория, содержащая основные файлы проекта.
 - settings.py: файл, содержащий настройки проекта.
 - urls.py: файл, содержащий маршруты URL-адресов проекта.
 - wsgi.py: файл, используемый сервером приложений для запуска проекта.
- Приложение/: директория, содержащая файлы и настройки приложения.
 - migrations/: директория, содержащая миграции базы данных приложения.
 - templates/: директория, содержащая шаблоны HTML-страниц приложения.
 - models.py: файл, содержащий модели приложения.
 - views.py: файл, содержащий представления приложения.
 - urls.py: файл, содержащий маршруты URL-адресов приложения.

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 34 |

ЗАКЛЮЧЕНИЕ

В заключение следует отметить, что разработка системы управления запасами для склада была успешно завершена. Система была разработана для повышения эффективности комплектации и маршрутизации продукции на складе. Цель проекта заключалась в разработке системы, которая позволила бы автоматизировать процесс отслеживания запасов, повысить точность управления запасами и сократить время, необходимое для выполнения задач, связанных с инвентаризацией.

В рамках данной курсовой работы были разработаны математическое, программное, техническое, лингвистическое, методическое и информационное обеспечения после проведения анализа существующих решений. Также были созданы сервер, сайт сформированы базы данных. Реализованы механизмы защиты информации в виде аутентификации и авторизации на сайте.

| | | | | | | |
|------|------|----------|-------|------|-----------------------------------|------|
| | | | | | <i>ТГТУ.09.03.01.025 КР ТЭ-ПЗ</i> | Лист |
| | | | | | | |
| Изм. | Лист | № докум. | Подп. | Дата | | 35 |

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

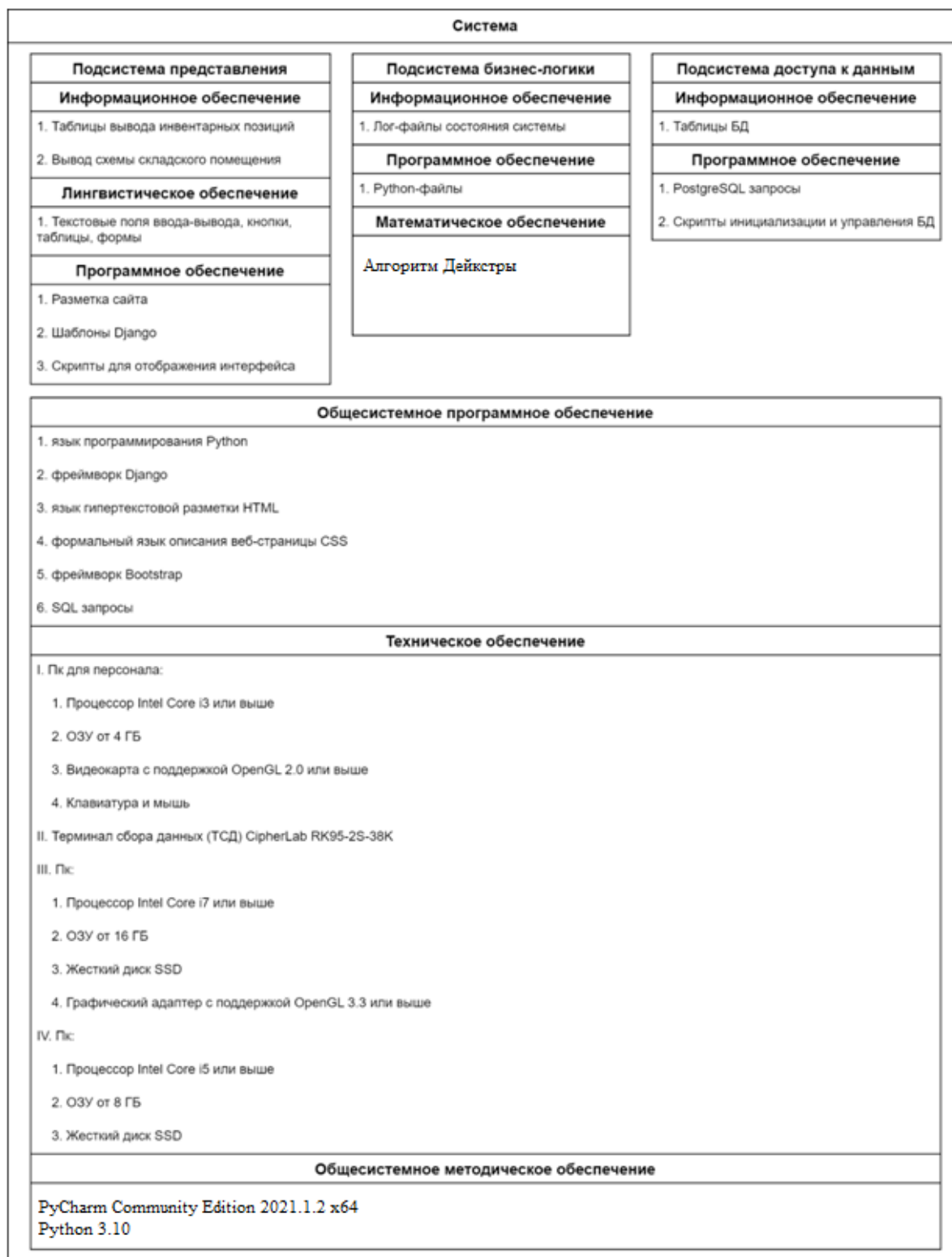
1. Python 3.11.3 official documentation [Электронный ресурс] // Python Software Foundation. – URL: <https://docs.python.org/3/>
2. PostgreSQL: The World's Most Advanced Open Source Relational Database [Электронный ресурс] // PostgreSQL – URL: <https://www.postgresql.org/>
3. Inventory management software [Электронный ресурс] свободная энциклопедия Wikipedia – URL: https://en.wikipedia.org/wiki/Inventory_management_software
4. Inventory Management System [Электронный ресурс] // ScienceDirect – URL: <https://www.sciencedirect.com/topics/economics-econometrics-and-finance/inventory-management-system>
5. Warehouse Management: A Complete Guide [Электронный ресурс] // ScienceDirect – URL: <https://www.sciencedirect.com/science/article/pii/S1877050917313120>

| | | | | | | |
|------|------|----------|-------|------|----------------------------|------|
| | | | | | ТГТУ.09.03.01.025 КР ТЭ-ПЗ | Лист |
| | | | | | | 36 |
| Изм. | Лист | № докум. | Подп. | Дата | | |

ПРИЛОЖЕНИЕ А

(обязательное)

Структурная схема



ПРИЛОЖЕНИЕ Б
(обязательное)
Функциональная схема

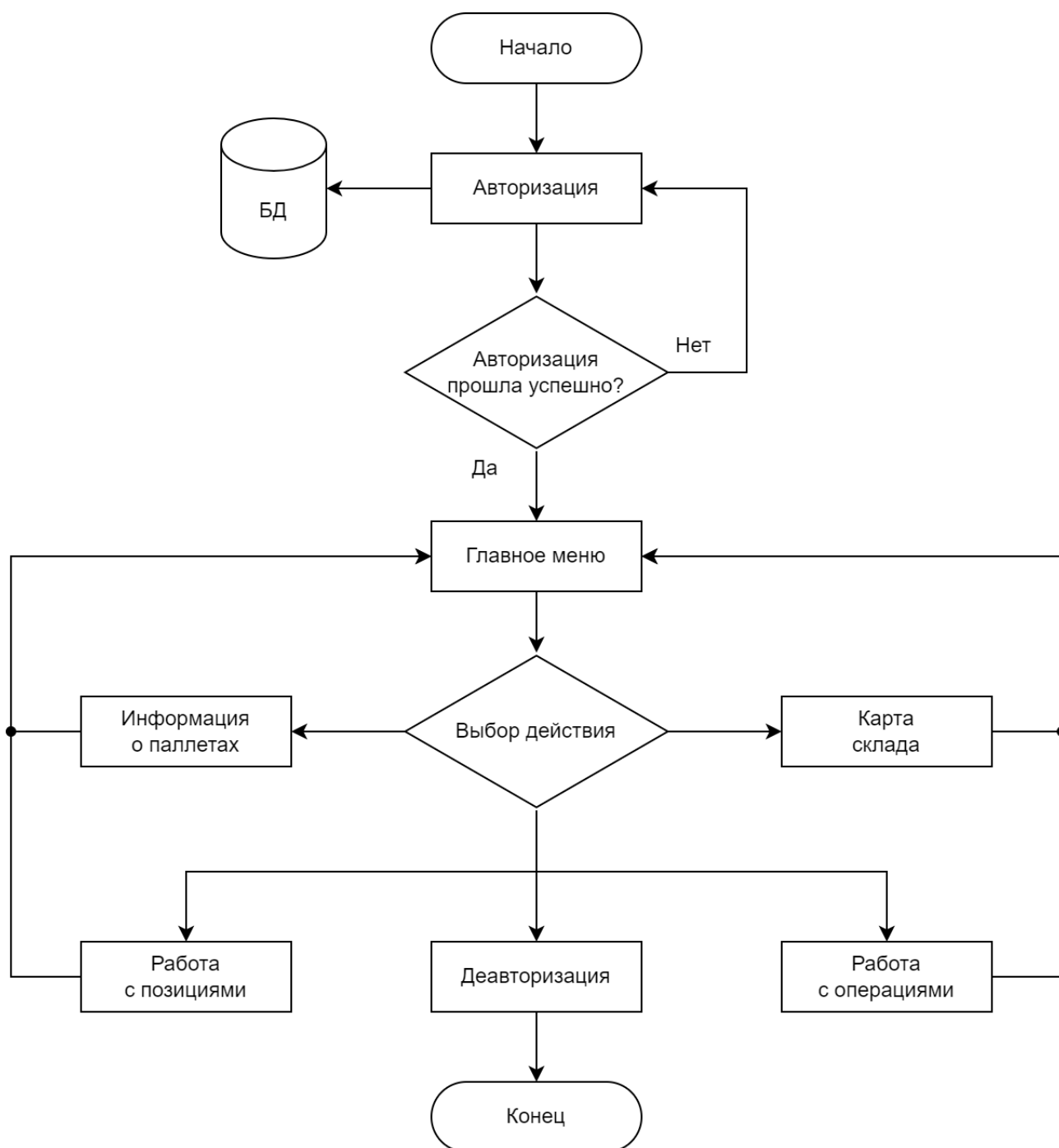


Рисунок Б.1. Веб-интерфейс пользователя. Общая схема работы

ПРИЛОЖЕНИЕ В
(обязательное)

Даталогическая схема

