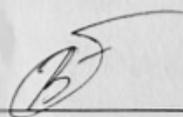


МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

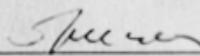
**КУРСОВОЕ ПРОЕКТИРОВАНИЕ**  
Проектирование приложения для автоматизации  
учета на складе предприятия  
Б-ПИ-21-01-161584. 8850-с. 05.000. КП.00

Студент  
гр. БПИ-21-МП1



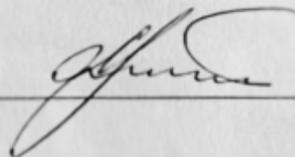
Н.С. Витюгова

Руководитель,  
профессор,  
доцент, доктор техн. наук



В.М. Гриняк

Консультант,  
старший преподаватель



Е.Г. Лаврушина

Владивосток 2025

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВВГУ»)

Институт информационных технологий и анализа данных  
Кафедра информационных технологий и систем

Индивидуальное задание  
на курсовое проектирование

Студенту гр. БПИ-21-МП1 Витюговой Наталье Сергеевне

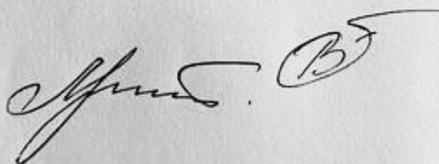
- 1 Описать текущее положение предприятия ООО «Сервиском»
- 2 Определить архитектуру системы
- 3 Сформулировать технические требования
- 4 Спроектировать прототип решения

Срок сдачи работы: 14.01.2025

Консультант,  
старший преподаватель

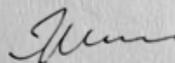
Лаврушина Е.Г.

Задание получил:



Витюгова Н.С.

Руководитель,  
профессор,  
доцент, доктор техн. наук



Гриняк В.М.

## Аннотация

Курсовая работа посвящена проектированию и разработке приложения для автоматизации учета на складе предприятия. Основной задачей проекта является повышение эффективности складских операций за счет сокращения временных затрат, а также минимизации вероятности ошибок при обработке документации и внесении данных.

В рамках курсовой работы была разработана концепция двухкомпонентной системы, включающей мобильное приложение для оперативной работы сотрудников склада и десктопную версию для анализа данных и выполнения управленческих функций, таких как сортировка и печать.

Для достижения поставленных целей проведен анализ текущего состояния предприятия, изучен существующий процесс складского учета и выявлены ключевые проблемы, которые препятствуют их эффективной работе. На основе полученных данных были сформулированы технические требования, разработаны прототипы интерфейсов, а также выбран технологический стек, включающий OpenServer для серверной части, SQLite для хранения данных, Kotlin для мобильного приложения и PyQt для ПК-версии.

Особое внимание в проекте было уделено разработке архитектуры системы. Были продуманы механизмы синхронизации данных между компонентами через локальную сеть, что обеспечивает бесперебойное взаимодействие между мобильным приложением и десктопной версией. Также разработан и внедрен механизм аутентификации пользователей для повышения безопасности системы и предотвращения несанкционированного доступа к данным. Управление данными организовано таким образом, чтобы максимально упростить выполнение ключевых операций.

Реализованные прототипы интерфейсов ориентированы на выполнение основных пользовательских сценариев. Для мобильного приложения это включает отображение накладных и внесение данных в систему. В десктопной версии реализованы функциональные возможности выведения списка всех накладных, их сортировки, редактирование и печати полученного документа.

Система разработана с учетом принципов простоты и интуитивной понятности интерфейсов, что позволяет минимизировать время на обучение персонала и сохранить преемственность с текущими методами работы предприятия. Предложенное решение не только оптимизирует учетные процессы, но и повышает общую производительность сотрудников, снижая вероятность ошибок при работе с данными.

## Содержание

Введение.....	5
1 Текущее состояние и ожидаемые результаты проектирования .....	6
2 Требования к разработке .....	8
2.1 Структура системы .....	8
2.2 Описание функциональных возможностей приложений .....	11
2.3 Технические требования.....	14
3 Особенности разработки.....	15
3.1 Технологический стек разработки .....	15
3.2 Выбор механизма аутентификации.....	19
3.3 Особенности сохранения данных.....	22
3.4 Обоснование ограничения отображения документов .....	23
3.5 Очищение баз данных.....	23
4 Макет приложений.....	25
4.1 UserFlow приложений.....	25
4.2 Прототип мобильного приложения.....	28
4.3 Прототип ПК-версии .....	33
Заключение.....	36
Список используемых источников.....	37
Приложение А.....	38

## Введение

Данная работа посвящена проектированию и разработке приложения для автоматизации учета на складе предприятия. В процессе реализации проекта была поставлена задача создать двухкомпонентную систему, включающую мобильное приложение для оперативной работы сотрудников склада и настольное приложение для анализа данных и взаимодействия с системой учета предприятия.

Актуальность разработки данного проекта обусловлена возросшей потребностью предприятия в повышении эффективности складских процессов, которые сопровождаются большим объемом ручного труда, для снижения вероятности ошибок при переносе данных из одних документов в другие. Современные методы учета, такие как бумажные накладные, требуют значительных временных и трудовых затрат, что приводит к снижению производительности и задержкам в документообороте. Такие ограничения снижают производительность труда и могут привести к задержкам в документообороте, что отрицательно влияет на эффективность компании.

В ходе выполнения работ проведен анализ текущего состояния предприятия, сформулированы технические требования к системе, а также выбраны технологии и подходы, которые позволяют решить задачу с учетом ограничений и условий бизнес-процессов компании. Одним из приоритетов стало создание эффективной системы синхронизации данных между мобильной и ПК-версией приложений, что способствует повышению удобства использования системы. Дополнительно был сделан акцент на улучшении интерфейсов для пользователей, чтобы обеспечить взаимодействие процессов с помощью системы обеспечения понятности и максимальной комфорта.

Проектируемое решение предусматривает выполнение широкого спектра действий, включая ввод данных, их надежное хранение, редактирование информации о статусе выполнения заказов, а также возможность печати данных для их последующего архивирования. Эти функции направлены на оптимизацию учета операций и повышение скорости обработки данных, что в конечном итоге обеспечивает общую эффективность работы предприятия.

## 1 Текущее состояние и ожидаемые результаты проектирования

На данный момент сотрудники в течение рабочего дня получают накладные от покупателей, в которых содержится информация о приобретенных товарах. Задача кладовщика состоит в обеспечении своевременной отгрузки заказанного товара. Используя предоставленные документы, сотрудник собирает необходимую продукцию на складе и делает пометки на копиях накладных. Эти пометки фиксируют количество фактически отгруженного товара, выраженного в тоннах и упаковках.

К концу рабочего дня сотрудник обязан собрать все накладные, проведенные за текущий день, и систематизировать данные. Эта работа включает перенос информации из всех накладных, проведенных за день, в специальную бумажную форму. В этой форме отображаются ключевые параметры: номер накладной, имя покупателя, категория и тип товара, а также количество отгруженного товара, измеренное в тоннах. Данный процесс позволяет структурировать информацию и упростить дальнейшую работу с ней.

После завершения заполнения бумажной формы необходимо перенести данные в программу 1С. Сотрудник вводит всю собранную информацию о количестве отгруженного товара, после чего программа автоматически генерирует уникальный «код расхода». Этот код является важным идентификатором и должен быть зафиксирован в бумажной форме, чтобы подтвердить завершение всех операций по данной накладной.

После завершения всех действий сотрудник должен сохранить документы, а именно – заполненную бумажную форму, в архиве предприятия. Архивация данных необходима для обеспечения их сохранности, возможности последующего анализа и выполнения требований регуляторов, связанных с учетом и документооборотом.

Проектируемое решение оптимизирует нынешний процесс. В течение рабочего дня сотрудники получают накладные, которые необходимо обработать. В мобильном приложении, установленном на их смартфонах, отображается полный список заказов с детальной информацией о количестве и типе товара, который требуется отгрузить. Сотрудники могут оперативно вносить изменения в данные, фиксируя объем выполненной отгрузки по каждой накладной. Это обеспечивает актуальность и точность учета в режиме реального времени.

К концу рабочего дня, после завершения всех отгрузок, сотрудник сохраняет внесенные данные в мобильном приложении. Приложение автоматически синхронизируется с десктопной версией системы через локальную сеть. Десктопное приложение предоставляет сотрудникам возможность детального анализа статистики за день: сколько единиц продукции, какого типа и кому было отгружено.

Одной из ключевых функций десктопной версии является возможность удобного копирования данных о количестве отгруженного товара (в тоннах и упаковках) для последующего внесения в общую систему учета 1С [1]. Это соответствует требованиям предприятия по документообороту. После ввода данных в систему 1С она автоматически генерирует уникальный «код расхода», который привязывается к каждому покупателю за текущий день. Сотрудник вносит этот код в таблицу десктопного приложения, а затем распечатывает итоговый документ, который сохраняется в архиве предприятия.

Предлагаемое решение избавляет сотрудников от необходимости постоянно носить бумажные документы и исключая ошибки, возникающие при ручном переносе данных из бумажных носителей в систему 1С. При этом сохраняется возможность архивирования необходимой документации в бумажном формате, что соответствует требованиям предприятия.

Важно подчеркнуть, что разрабатываемое решение не будет напрямую интегрироваться с базой данных платформы 1С. Это связано с тем, что технический отдел предприятия не дал разрешения на внесение изменений в текущую конфигурацию платформы, а добавление дополнительного элемента в решение увеличит количество действий в работе бухгалтерии.

После завершения разработки предложенного решения проект будет представлен на согласование директору предприятия. В дальнейшем сотрудники технического отдела самостоятельно реализуют интеграцию между бухгалтерской системой и мобильным приложением. Это позволит обеспечить отображение актуальных накладных в приложении без необходимости изменения структуры существующей базы данных.

Подобный подход гарантирует сохранение стабильности и безопасности работы платформы 1С, а также упрощает процесс внедрения нового решения, поскольку оно будет максимально автономным от текущей учетной системы.

Решение будет разрабатываться с нуля, так как ранее данная проблема не поднималась, и готовых решений для ее устранения не существует.

ЕРС – Event-driven Process Chain, Событийная цепочка процессов. ЕРС предназначена для описания порядка выполнения БП в виде последовательности действий, управляемых событиями и выполняемых исполнителями [2]. Нотации ЕРС процессов до проектирования и после представлены в приложении А на рис. А.1 и рис. А.2 соответственно.

## 2 Требования к разработке

### 2.1 Структура системы

Система данной разработки будет состоять из двух компонентов: мобильное приложение для учета и ПК-версия для отображения данных. Блок-схема алгоритма работы изображена на рисунке 1.

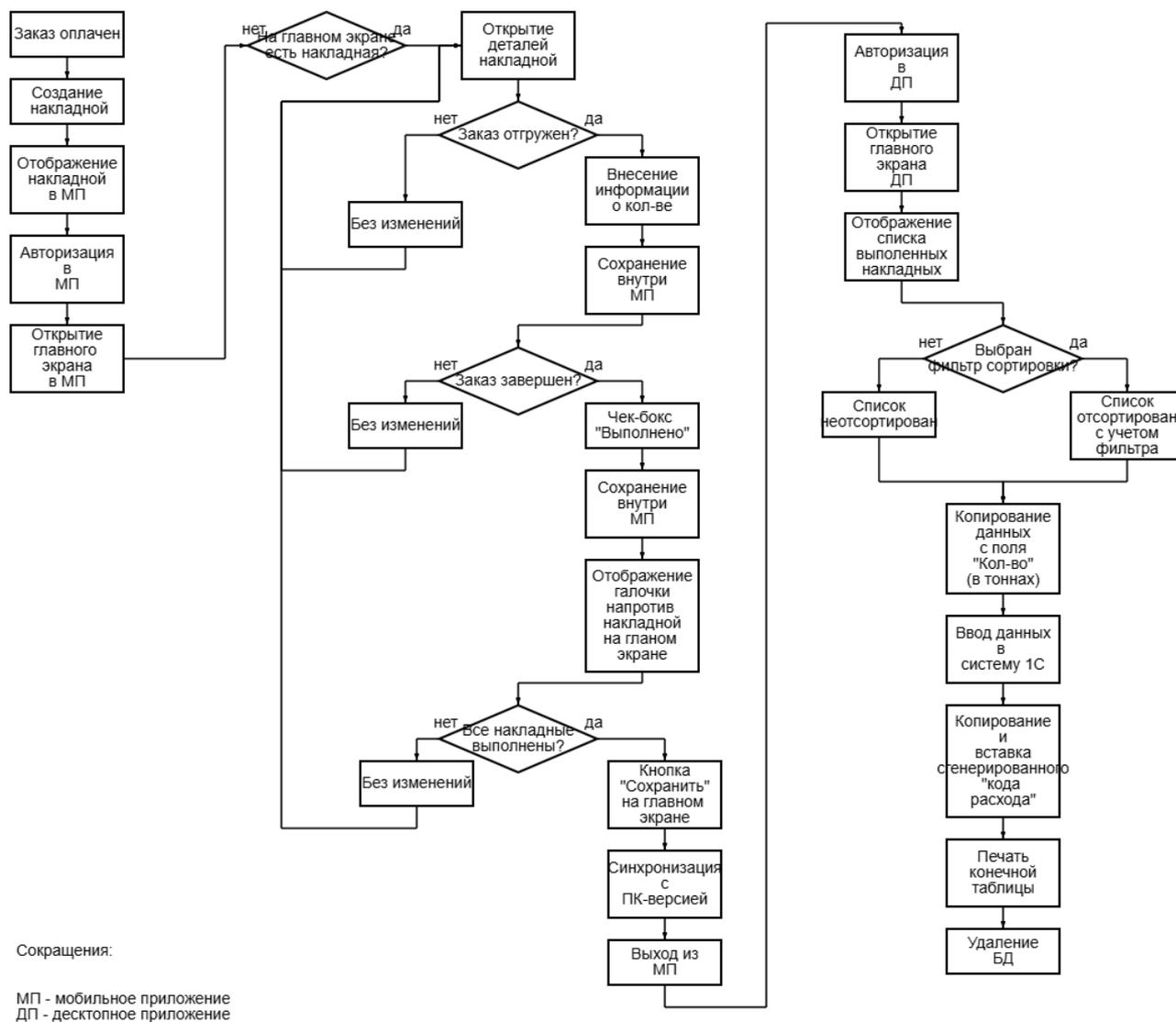


Рисунок 1 – Блок-схема алгоритма работы

Решение не будет интегрироваться в систему 1С студентом, поскольку таковы были условия предприятия. Необходимо рассмотреть, какие сложности будет иметь данная разработка.

Серверной части должна обеспечить надежное хранение данных, чтобы они были доступны как с мобильных устройств, так и с ПК. Для этого приложение должно принимать

данные с мобильного устройства, хранить их в базе данных и обеспечивать их отображение в ПК-версии.

Передача данных между мобильным приложением и ПК-версией может быть организована через облачную серверную инфраструктуру или локальную сеть.

Варианты решения:

Первое решение – облачный сервер. Облачные серверы — это не физические, а виртуальные серверы, работающие в среде облачных вычислений, к которым неограниченное количество пользователей может получить доступ по запросу [3]. Это наиболее универсальный способ, который не требует, чтобы мобильные устройства и ПК находились в одной сети. Сотрудники вводят данные в приложении. После нажатия кнопки "Сохранить", данные отправляются на облачный сервер через интернет. Серверная часть получает данные от мобильного приложения, сохраняет их в базе данных и обеспечивает доступ к данным через веб-интерфейс или ПК-программу. ПК-версия обращается к серверу за данными в реальном времени или по запросу. Далее система отображает данные в таблице, позволяя пользователю копировать их для работы с 1С.

Второе решение – локальная сеть. Локальная вычислительная сеть (ЛВС, локальная сеть; англ. Local Area Network, LAN) — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт) [4]. Этот подход подходит, если необходима локальная структура без интернета. Мобильное приложение и ПК-версия подключаются к одному серверу, расположенному в локальной сети, данные передаются через API-соединение, аналогичное облачному варианту, но внутри корпоративной сети. API (англ. application programming interface — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными [5]. ПК-версия запрашивает данные из локального сервера и отображает их.

Третье решение – прямой обмен через файлы. Если структура системы минимальна, то возможна передача данных через файлы, например, CSV или Excel. В этом случае мобильное приложение генерирует файл с данными. Сотрудник отправляет файл на ПК через Bluetooth, локальный Wi-Fi или облачное хранилище, а ПК-версия загружает файл и отображает данные.

Рассмотрим таблицу (таблица 1) с плюсами и минусами каждого варианта.

Таблица 1 – Сравнительный анализ вариантов структуры системы

	Плюсы	Минусы
Облачный сервер	Доступ к данным из любой точки с интернетом.	Зависимость от стабильного интернет-соединения.
	Легко добавить новых пользователей и функции.	Требуются затраты на аренду облачного хранилища и разработку серверной части
	Автоматическая синхронизация между устройствами.	Необходимость настройки безопасности для защиты данных (шифрование, авторизация).
Локальная сеть	Не требуется интернет-соединение.	Данные доступны только в пределах локальной сети.
	Данные остаются внутри инфраструктуры предприятия.	Требуются установка и обслуживание локального сервера.
	Минимальные затраты на хостинг и облачную инфраструктуру.	
	Высокая скорость передачи данных внутри сети.	
Прямой обмен	Не требуется сервер или сложная архитектура.	Много ручных операций: передача, загрузка и обработка файлов.
	Не требует постоянного интернет-соединения.	Ошибки возможны при ручном переносе файлов между устройствами.
	Минимальные затраты на разработку и поддержку.	Нет автоматической синхронизации между устройствами.

Опираясь на таблицу, наиболее подходящим вариантом будет решение, основанное на локальной сети.

## 2.2 Описание функциональных возможностей приложений

Реализация передачи данных между мобильным приложением и ПК-версией через локальную сеть требует включает несколько ключевых этапов: настройку серверной части и разработку приложений. Этот процесс должен быть организован таким образом, чтобы система была надежной, удобной для использования и легко поддерживалась.

Первым шагом является выбор технологии, на основе которой будет работать локальная сеть. Основной задачей на этом этапе является определение, где будет размещаться серверная часть. Возможные варианты:

- локальный сервер;
- мини-сервер;
- роутер.

Наиболее оптимальным решением является развертывание локального сервера на компьютере предприятия. Это позволит эффективно использовать уже имеющиеся ресурсы, что важно в условиях ограниченного бюджета разработки. Кроме того, локальный сервер обеспечит стабильное и надежное соединение между устройствами.

Следующим ключевым этапом разработки является проектирование архитектуры системы, которая объединяет мобильное приложение, ПК-версию и серверную часть. Каждый компонент должен быть четко интегрирован и выполнять свою роль, чтобы обеспечить надежную и удобную работу системы.

Серверная часть является центральным узлом, обеспечивающим управление базой данных, обработку запросов от клиентов (мобильного приложения и ПК-версии) и синхронизацию данных между ними. Сервер должен быть оснащен функционалом для безопасного хранения данных, обработки информации и обеспечения стабильности соединения.

Мобильное приложение выполняет роль инструмента для непосредственного ввода данных сотрудниками. Решение должно реализовывать функции отображения всех накладных в приложении телефона. Сотрудник будет иметь возможность регулировать все бумаги в приложении.

Спроектированное решение будет позволять кладовщику вносить изменения в течение дня. В конце рабочего дня, после сохранения, сотрудник сможет просмотреть статистику в ПК-версии аналогичную той, что до этого вел самостоятельно в бумажном варианте.

Сотрудник должен будет скопировать данные и внести их в систему 1С, после чего, скопировать сгенерированный код, внести его в список накладных, и распечатать, если это нужно будет для архива предприятия. Его функционал должен включать:

- отображение накладных. Приложение должно предоставлять сотруднику доступ к списку накладных, созданных в течение рабочего дня;
- внесение данных об отгрузке. Пользователь должен иметь возможность указывать количество отгруженного товара относительно каждой накладной, как в упаковках, так и в тоннах;
- поиск по накладным. Функция поиска упрощает работу с большим количеством записей, предоставляя быстрый доступ к нужным документам;
- синхронизация данных с ПК-версией. Обеспечивается через кнопку «Сохранить», которая инициирует процесс передачи данных на сервер;
- сохранение данных внутри приложения. Это гарантирует, что данные сохраняются локально на устройстве до их синхронизации с сервером.

ПК-версия системы предназначена для более детального анализа, редактирования и архивирования данных. Среди ключевых функций должны быть:

- просмотр списка накладных. Пользователь получает возможность видеть полный перечень всех накладных, которые были проведены за текущий день;
- сортировка отгруженного товара. Данные должны быть представлены в разрезе организаций, что позволяет быстро анализировать выполненные заказы;
- редактирование записей. Сотрудник может вносить изменения в определенные поля, например, добавлять данные, сгенерированные в 1С, или корректировать записи;
- печать списка накладных. В конце рабочего дня сотрудник должен иметь возможность распечатать весь список для хранения в бумажном архиве.

Для каждого элемента разрабатываемого решения необходимо выбрать стек используемых программ, отталкиваясь от требований к ПО.

На завершающих этапах необходимо провести тестирование для проверки корректности передачи данных между мобильным приложением, сервером и ПК. Основные задачами тестирования будут:

- проверка стабильности системы в условиях реального использования;
- тестирование высокой нагрузки и потери соединения, чтобы убедиться в надежности системы;
- обнаружение и устранение багов, связанных с синхронизацией данных.

Финальный этап — подготовка подробных инструкции по использованию мобильного приложения и ПК-версии. Эти материалы должны включать в себя руководство по использованию основными функциями.

Диаграмма Ганта – это график работ по проекту, который строится в виде таблицы с этапами и ответственными за их выполнение. Диаграмма состоит из двух осей: вертикальной со списком задач и горизонтальной со сроками [6]. Основные этапы изображены на рисунке 2.

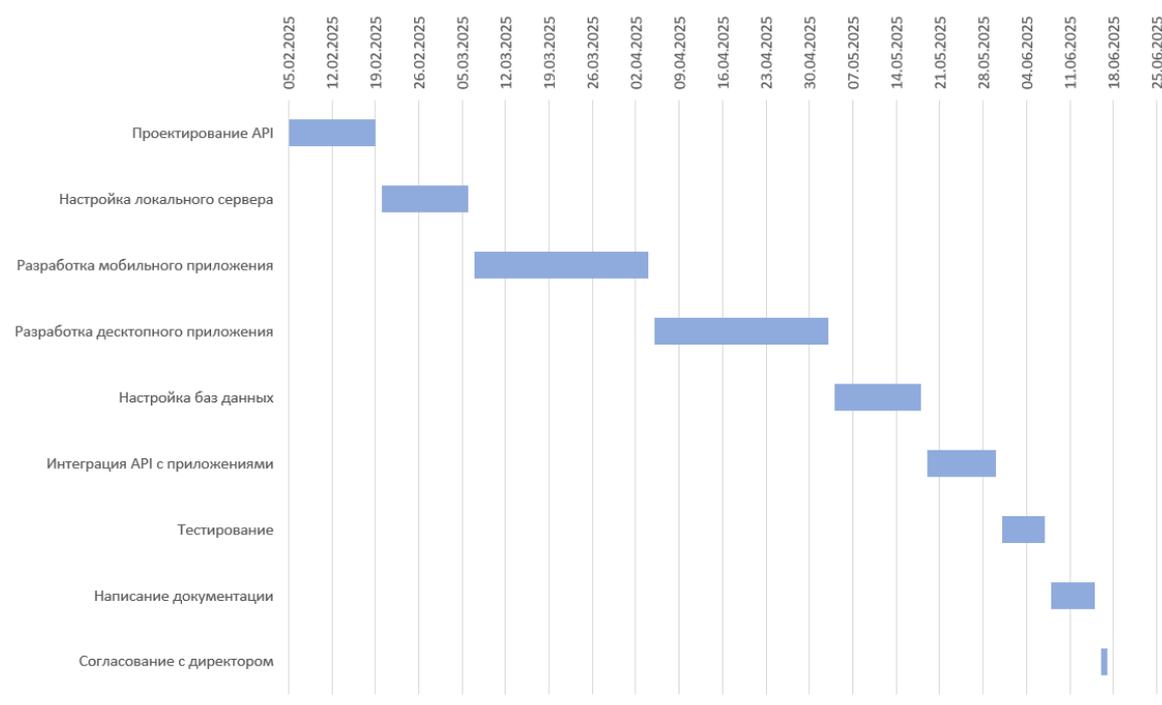


Рисунок 2 – Диаграмма Ганта

На представленной диаграмме видно время, отведенное на выполнение основных этапов проекта. Анализируя временные рамки, можно отметить, что наиболее длительными и ресурсоемкими этапами являются разработки мобильного приложения и создания десктопной версии программы. На выполнение каждой из этих задач выделяется по 4 недели.

На втором месте по устойчивости задачи мониторинга, связанные с проектированием API, настройкой локального сервера и конфигурацией базы данных. На выполнение каждого из этих этапов отведено по 2 недели.

Продолжительность каждого этапа обуславливает их необходимость тщательной разработки для обеспечения стабильности, надежности и удобства будущей системы.

## 2.3 Технические требования

Для корректной работы проектируемого решения необходимо, чтобы разрабатываемые программы могли работать с учетом некоторых вводных требований. Рассмотрим подробнее.

Мобильное приложение должно быть совместимо с устройствами под управлением выключателей системы Android версии 8.0 (Oreo) или выше. Минимальный объем оперативной памяти устройства должен составлять не менее 2 ГБ. Это необходимо для стабильной работы приложения даже при запуске нескольких фоновых процессов. Для установки и работы приложения требуется минимум 100 МБ свободного места в хранилище устройства. Приложение должно иметь возможность подключения к локальной сети через Wi-Fi.

Программа для ПК-версии должна быть совместима с системой Windows, начиная с версии Windows 10 (64-разрядная). Рекомендуется наличие двухъядерного процессора с тактовой частотой не менее 2 ГГц. Минимальный объем оперативной памяти должен составлять 4 ГБ, чтобы программа могла работать без задержек и обеспечивать комфортное использование. Программе потребуется от 200 до 500 МБ свободного места. Необходимо предусмотреть возможность подключения к локальной сети через Wi-Fi или посредством локального соединения с сервером.

Локальный сервер должен функционировать под управлением устройств системы Windows 10 (64-разрядная). Для работы сервера необходим двухъядерный процессор с частотой не менее 2 ГГц. Требуется наличие 4 ГБ оперативной памяти, что позволит серверу обрабатывать данные и обеспечивать выполнение запроса. На диске должно быть минимум 700 МБ свободного места, необходимого для установки Open Server, базы данных и других серверных компонентов. Сервер должен поддерживать подключение через Ethernet или Wi-Fi для обеспечения стабильного и надежного соединения с локальной сетью.

Все разрабатываемые приложения должны строго соответствовать установленным и подробно описанным техническим параметрам, которые были сформулированы в соответствии с требованиями проекта, либо превосходить их, чтобы обеспечить более высокий уровень функциональности, надежности и удобства использования.

## 3 Особенности разработки

### 3.1 Технологический стек разработки

Для выбора программ необходимо ввести критерии. Среди таковых должны быть: легкость настройки, производительность, кроссплатформенность, поддержка API, сообщество и документация.

Для серверной части будем использовать локальный сервер. Локальный сервер – программное обеспечение, позволяющее сделать отладку сайта на локальном (домашнем) компьютере, без выхода в интернет. Локальный сервер полностью имитируют работу хостинга и имеет аналогичный функционал. [7]

Среди локальных серверов есть много вариантов, например, XAMPP, Denwer [8] и Open Server [9]. Рассмотрим таблицу сравнения данных вариантов (таблица 2).

Таблица 2 – Сравнительный анализ локальных серверов

Критерий	XAMPP	Denwer	Open Server
Легкость настройки	Легкая установка.	Простая установка и удаление программы.	Имеет приятный и дружелюбный интерфейс, прост в установке.
Производительность	Умеренная. Более медленный в работе, чем, OpenServer.	Умеренная. Отсутствие обновлений. Последняя версия программы была выпущена в 2013 году.	Хорошая для небольших и средних проектов.
Кроссплатформенность	Windows (7 и выше), Linux, macOS.	Только Windows (XP и выше).	Только Windows (7 и выше).
Поддержка API	Средняя. Требуется настройка PHP или интеграция с Python.	Средняя. Больше подходит для статических сайтов или PHP-скриптов.	Хорошая. Поддерживает PHP и Python, легко интегрируется с фреймворками.

Сообщество документация	и	Большое сообщество.	Устаревшее сообщество.	Большое сообщество.
----------------------------	---	------------------------	---------------------------	------------------------

Анализируя возможности различных серверных решений, можно уверенно сказать, что Denwer не является подходящим вариантом для разработки. Этот инструмент, несмотря на свою популярность в прошлом, сегодня значительно уступает современным решениям из-за использования устаревших технологий и недостаточной производительности.

В нашем случае наиболее предпочтительным выбором является OpenServer. Это программное обеспечение выделяется своей простотой в установке и настройке, а также высокой степенью гибкости. OpenServer поддерживает широкий спектр современных технологий, включая PHP, MySQL и другие инструменты, что делает его универсальным решением для разработки.

OpenServer поддерживает широкий выбор СУБД. Помимо этого, у ПО есть предустановленные СУБД. Система управления базами данных, сокр. СУБД (англ. Database Management System, сокр. DBMS) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных (БД) [10]. Среди них находятся: MySQL, MariaDB, SQLite, PostgreSQL. SQLite будет идеальным решением для небольших приложений. С учетом функционала разрабатываемого проекта можно использовать именно SQLite. В случае, если в процессе разработки станет понятно, что возможностей данной варианта недостаточно, то можно использовать MySQL, поскольку эта СУБД наиболее популярная и универсальная.

Поскольку мобильное приложение будет работать на платформе Android, то язык программирования будет выбираться из Kotlin и Java.

Java является классическим выбором. Этот язык программирования хоть и устаревающий, но используется в большинстве существующих приложений, что дает большую базу документацию с примерами решений и огромное сообщество. Java имеет высокую производительность, однако, вместе с этим содержит более строгий синтаксис, что может вызвать трудности у новичков.

Kotlin – основной язык разработки Android после 2017 года. Он подходит для новых проектов благодаря современному синтаксису, который ускоряет разработку. Несмотря на простоту, Kotlin работает с ныне существующими проектами, работающими на Java, и полностью совместим с Java – библиотеками. Благодаря данным преимуществам Kotlin имеет быстрорастущее сообщество и хорошую документацию.

Kotlin и Java взаимозаменяемые языки программирования. Решающим фактором в выборе языка будет простота и легкость синтаксиса, в чем преуспевает Kotlin, поэтому для разработки мобильного приложения будет использоваться именно он.

Выбор языка программирования для разработки ПК-версии приложения будет зависеть от нескольких критериев. Среди них: производительность, простота разработки, легкость настройки, ресурсозатратность, документация. Выбора языка программирования будет производиться из Electron.js, PyQt/PySide (Python) и .NET (C#), опираясь на таблицу со сравнением (таблица 3).

Таблица 3 – Сравнительный анализ языков программирования

	Electron.js	PyQt/PySide (Python)	.NET (C#)
Производительность	Требует больше ресурсов из-за использования Node.js и Chromium.	Высокая, ближе к нативной.	Высокая, особенно на Windows, так как .NET оптимизирован для работы с Windows API.
Простота разработки	Требует знаний HTML, CSS и JavaScript.	Требует знаний Python и особенностей Qt.	Требует знаний C# и концепций Windows-программирования.
Легкость настройки	Проста в установке и настройке, особенно для веб-разработчиков.	Настройка требует установки Python и библиотек Qt.	Интеграция с Visual Studio делает настройку относительно простой.
Ресурсозатратность	Относительно большой из-за Chromium (примерно 50-100 МБ минимально).	Меньше, чем у Electron, но зависит от дополнительных библиотек.	Меньше, чем у Electron, особенно при использовании .NET Core.
Документация	Обширное сообщество, богатая	Большое сообщество Python-разработчиков,	Обширное сообщество Microsoft, богатая

	документация и примеры.	хорошая документация Qt.	документация и официальная поддержка.
--	----------------------------	-----------------------------	---

Опираясь на таблицу, можно сделать вывод, что наиболее подходящими вариантами будут PyQt/PySide и .NET (C#). Оба варианта будут предпочтительнее, но, так как приложение будет небольшое, но функциональное, то выбор падает на PyQt/PySide.

### 3.2 Выбор механизма аутентификации

Аутентификация (англ. authentication «реальный, подлинный») — процедура проверки подлинности, например, проверка подлинности пользователя путём сравнения введённого им пароля (для указанного логина с паролем), сохранённым в базе данных пользовательских логинов. [11]

Аутентификация пользователя представляет собой ключевой процесс, обеспечивающий проверку введенных данных с последующим предоставлением доступа к функциональной системе, если информация была введена правильно. Этот процесс играет важную роль в защите данных и сохранении конфиденциальности информации, особенно в сфере коммуникаций, связи с управлением деятельностью или с учетом интересов предприятия. Для реализации привлечения сотрудников к учетной системе могут быть предложены два основных сценария, каждый из которых обладает преимуществами и достижениями, которые необходимо учитывать при выборе подходящего метода.

Первый сценарий аутентификации представляет собой способ, при котором учетные записи сотрудников создаются непосредственно администрацией системы. Такой подход предполагает централизованный контроль за процессом регистрации пользователей и их доступом к системе. Учетные записи могут быть созданы автоматически во время развертывания системы, когда администрация заранее вводит данные о сотрудниках в базу. Данные для входа, такие как логин и пароль, предоставляются сотруднику администрацией.

Для повышения безопасности можно внедрить механизм регулярной смены паролей. Например, сотрудникам будет предлагаться обновлять пароль каждые 60 или 90 дней. Это позволит предотвратить возможную утечку данных.

Второй сценарий – это самостоятельная регистрация учетной записи пользователем. В рамках этого подхода создается форма для самостоятельной регистрации, где сотрудники вводят свои данные, задают логин и пароль, а также могут указать дополнительную информацию для восстановления доступа, например, электронную почту или контрольный вопрос.

Рассмотрим плюсы и минусы каждого сценария в таблице (таблица 4).

	Плюсы	Минусы
Сценарий 1	Удобство для сотрудников, так как процесс регистрации исключен.	При потере пароля сотруднику придется обращаться к техподдержке, что может занять время и вызвать задержки в работе.
	Централизованный контроль над учетными записями со стороны администрации системы.	Требуется дополнительное время на создание учетных записей для всех сотрудников перед запуском системы.
	Меньшая вероятность ошибок при создании учетной записи, так как этим занимается технический персонал.	
Сценарий 2	Снижение нагрузки на техническую поддержку, так как сотрудники самостоятельно регистрируют и восстанавливают свои учетные записи.	Возможность ошибок со стороны пользователя при регистрации, например, слабый пароль.
	Ускорение процесса входа в систему, так как регистрация не зависит от действий технического персонала.	Риск утраты доступа при отсутствии надежной информации для восстановления (например, если пользователь забывает указать актуальную почту).

		Требуется дополнительная защита от автоматической регистрации и злоумышленников.
--	--	--

Таблица 4 – Сравнение сценариев

В рамках первого сценария аутентификации недостатком может быть высокая нагрузка на технический персонал, в тех случаях, когда сотрудники теряют свои учетные данные для входа в систему. Однако в текущих условиях предприятия этот минус не является критичным, так как на данном этапе численность сотрудников остается относительно небольшой. Это позволяет технической службе эффективно справляться с возможными запросами и обеспечивает стабильную работу системы. Если в будущем численность сотрудников на предприятии значительно возрастет, потребность в изменении подхода к аутентификации станет очевидной. В этом случае можно будет перейти на второй сценарий, предполагающий самостоятельную регистрацию учетных записей пользователями. Такой подход будет более гибким и снизит нагрузку на техническую поддержку, позволяя сотрудникам быстрее решать вопросы, связанные с регистрацией и восстановлением доступа.

Основываясь на сравнении двух сценариев, можно сделать вывод, что на текущем этапе развития предприятия первый сценарий выглядит наиболее подходящим и выгодным.

### 3.3 Особенности сохранения данных

Одним из ключевых вопросов при разработке приложения является вопрос сохранения данных. В данном случае вся информация о каждом заказе сохраняется автоматически непосредственно в мобильном приложении. После того как сотрудник вносит изменения в накладную, система запоминает эти данные, делая их доступными для дальнейшей работы в течение дня.

Сотрудник имеет возможность неоднократно вносить изменения в данные каждой позиции, редактируя информацию в реальном времени. Это обеспечивает удобство в случаях, когда объемы отгруженного товара или другие параметры заказа изменяются по ходу выполнения задач. В конце рабочего дня, когда все накладные будут обработаны и закрыты, сотруднику необходимо нажать специальную кнопку «Сохранить», которая находится в главном меню приложения. Эта кнопка инициирует процесс синхронизации данных с ПК-версией системы.

В случае, если сотрудник не нажмет кнопку «Сохранить», то внесенные данные останутся доступны только в мобильном приложении. В ПК-версии система не отобразит изменения, так как синхронизация не будет произведена.

Если синхронизация данных выполняется раньше, например, в середине рабочего дня, то в ПК-версии отобразятся только те накладные и данные, которые уже были внесены на момент нажатия кнопки. В таком случае информация по незаполненным позициям

останется пустой: ячейки, предназначенные для отображения количества отгруженного товара или других параметров, будут оставаться незаполненными.

Кнопка «Сохранить» выполняет передачу данных из мобильного приложения в ПК-версию, а также отвечает за их перезапись в случае изменения информации.

### 3.4 Обоснование ограничения отображения документов

Следующим важным аспектом, требующим анализа, является возможность просмотра накладных за предыдущие дни. Данный вопрос напрямую связан с функциональными обязанностями сотрудников и правилами работы предприятия, для которого разрабатывается приложение. Чтобы принять решение о необходимости внедрения такой функции, важно детально рассмотреть текущие процессы обработки заказов и требования, предъявляемые к работе персонала.

Согласно установленным правилам предприятия, кладовщик обязан завершить обработку всех заказов в тот же день, в который они были проведены в системе. Например, если покупатель оформил заказ 12-го числа, то сотрудники склада должны полностью отгрузить заказ до конца рабочего дня 12-го числа. Правила предприятия не допускают растягивания выполнения заказов на несколько дней, чтобы поддерживать оперативность и строго соблюдать сроки.

Этот процесс регламентируется не только внутренними стандартами, но и логистическими требованиями компании. Следовательно, каждая накладная закрывается строго в день ее создания, и сотрудники обязаны завершить все действия, связанные с ее выполнением, до окончания рабочего дня.

В конце дня сотрудники формируют бумажный архив выполненных заказов, который сохраняется в соответствии с установленными процедурами учета. Это обеспечивает физическое подтверждение выполненных операций, которое может быть использовано для отчетности или проверки в будущем.

С учетом этих факторов необходимость в создании базы данных, которая будет хранить историю заказов за предыдущие дни, представляется избыточной. Данные, не относящиеся к текущему дню, не требуют обработки, так как работа с ними противоречит существующим правилам. Это упрощает разработку приложения, снижая объем хранения данных и повышая производительность системы.

### 3.5 Очищение баз данных

На основе анализа предыдущего вопроса следующим важным шагом является определение подхода к реализации процесса очистки базы данных. Эта функция должна

быть продумана таким образом, чтобы обеспечить сохранность данных до их обработки и избежать риска потери важной информации.

Одним из наиболее рациональных и удобных решений является привязка процесса удаления данных к завершению ключевых действий пользователя, таких как печать отчетов. В данном случае удаление базы данных в ПК-версии предлагается сделать доступным только после нажатия на кнопку «Печать». Такой подход позволяет гарантировать, что вся необходимая информация будет предварительно сохранена в виде физического или цифрового отчета. Подобный подход сводит к минимуму риск потери важных данных и повышению надежности системы.

Функционал удаления данных в ПК-версии должен быть реализован с учетом синхронизации с мобильными приложениями. Очистка базы данных на ПК должна автоматически удалять данные, хранящиеся в мобильной версии приложения. Такой подход обеспечивает полную целостность и согласованность данных в системе, обнаружение несоответствий или дублирование информации. Благодаря подобному решению также будет упрощена работа сотрудников, так как пользователям не придется выполнять дополнительные действия для удаления информации из мобильного приложения.

Отдельное внимание при разработке уделено предотвращению случайного удаления данных. В рамках данной концепции не предусмотрена отдельная кнопка для удаления информации в мобильном приложении. Такой подход позволяет не запустить случайно процесс обработки данных сотрудниками, которые используют мобильное устройство. Вместо этого управление процессом удаления централизовано в ПК-версии, что повышает общий уровень безопасности системы.

Для повышения надежности и предотвращения ошибок в интерфейс ПК-версии можно внедрить дополнительный механизм подтверждения при удалении данных. Например, после нажатия кнопки «Удалить базу» система может запросить подтверждение действия с указанием, что все данные будут очищены как на ПК, так и в мобильном приложении. Это добавляет еще один уровень защиты от случайных действий пользователя. Для усиления эффекта можно добавить визуальные составляющие, такие как цветное выделение текста.

## 4 Макет приложений

### 4.1 UserFlow приложений

UserFlow (англ. Пользовательский поток, или путь пользователя) – это схема движения пользователя, наглядный разветвленный сценарий его взаимодействия с конкретным цифровым продуктом: приложение или сайтом [12].

UserFlow представляет собой визуализацию точек входа, переходов и экранов, которые пользователь проходит на пути к достижению своей цели. Это наглядное отображение процесса выполнения определенной задачи или достижения пользовательской цели.

Дизайн интерфейса должен быть максимально привычным и интуитивно понятным для пользователя, чтобы исключить любое чувство дискомфорта или когнитивного диссонанса во время работы с приложением. Основной задачей является создание такой структуры интерфейса, которая будет логично восприниматься и не потребует дополнительного времени на адаптацию.

Для достижения этого результата в процессе разработки будет необходимо опираться на документации и формы, которые сотрудники использовали ранее в своей работе до внедрения проектируемого решения. Такой подход позволит сохранить преемственность между привычными методами работы и новыми цифровыми инструментами, что существенно повысит удобство использования приложения, а также ускорит процесс его внедрения в рабочие процессы.

UserFlow будет реализовываться в Miro [13]. Поскольку, разрабатываемое решение будет состоять из нескольких частей, рассмотрим UserFlow каждого в отдельности.

UserFlow мобильного приложения (рис. 3) начинается с экрана аутентификации. После успешной аутентификации пользователь попадает на главный экран. Этот экран представляет собой список всех актуальных накладных, доступных для обработки. Для облегчения работы предусмотрена функция поиска накладных, которая позволяет быстро найти нужный документ. Пользователь может воспользоваться поисковой строкой в верхней части экрана, введя ключевые данные.

После выбора нужной накладной пользователь переходит на следующий этап UserFlow — экран с описанием заказа. Этот экран предоставляет полный доступ ко всем деталям, связанным с заказом. Информация организована в виде структурированной таблицы. Этот экран дает возможность сотруднику получить полное представление о заказе. На экране с описанием заказа пользователь может не только просматривать данные, но и вносить изменения.

После фиксации данных сотрудник может отметить статус выполнения заказа, поставив галочку в чек-боксе «Выполнено». Эта отметка служит визуальным индикатором того, что накладная обработана полностью, что упрощает контроль за выполнением задач.

После внесения всех изменений пользователь может вернуться на главный экран приложения с помощью соответствующей кнопки.

Важным этапом UserFlow является синхронизация данных между мобильным приложением и ПК-версией системы. Изменения сохраняются автоматически внутри мобильного приложения, однако, их интеграция с ПК-версией требует нажатия кнопки «Сохранить» на главном экране.

Если сотрудник не нажмет кнопку сохранения, изменения останутся только в локальной версии приложения и не будут видны в системе на ПК.

На главном экране находится кнопка выхода из учетной записи. После выхода пользователь возвращается на экран аутентификации.

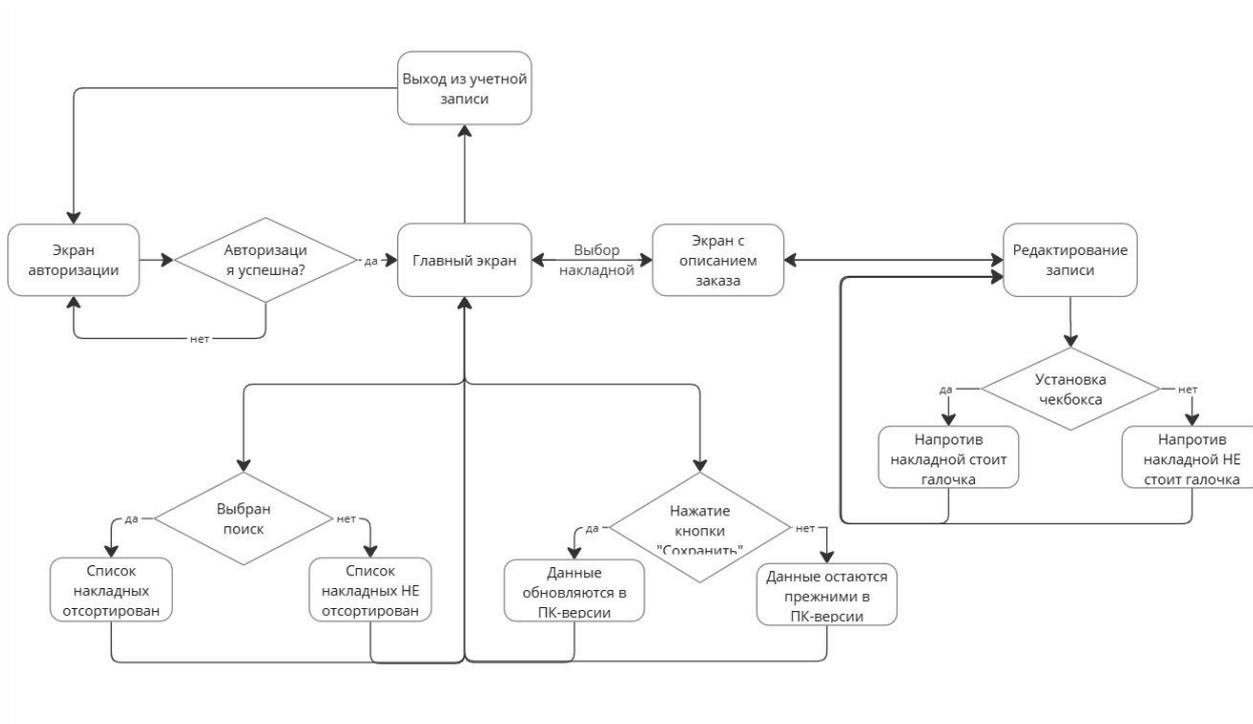


Рисунок 3 – UserFlow мобильного приложения

UserFlow ПК-версии (рис. 4) начинается с экрана аутентификации, где пользователь вводит свои учетные данные для доступа к системе. После успешного ввода данных пользователь получает доступ к основному функционалу приложения.

После входа в систему пользователь попадает на основной экран, где представлена таблица со списком накладных за текущий день. Для удобства работы предусмотрена возможность сортировки данных.

Одной из главных задач ПК-версии приложения является обработка данных по накладным. Пользователь может:

- копировать количество отгруженного товара для каждой позиции из таблицы;
- вносить сгенерированный код расхода, который привязывается к каждой накладной.

После внесения всех необходимых изменений сотрудник может сохранить данные в системе.

Когда все данные внесены и проверены, пользователь может распечатать таблицу с накладными за текущий день. Это действие выполняется для сохранения данных в архиве в физическом виде.

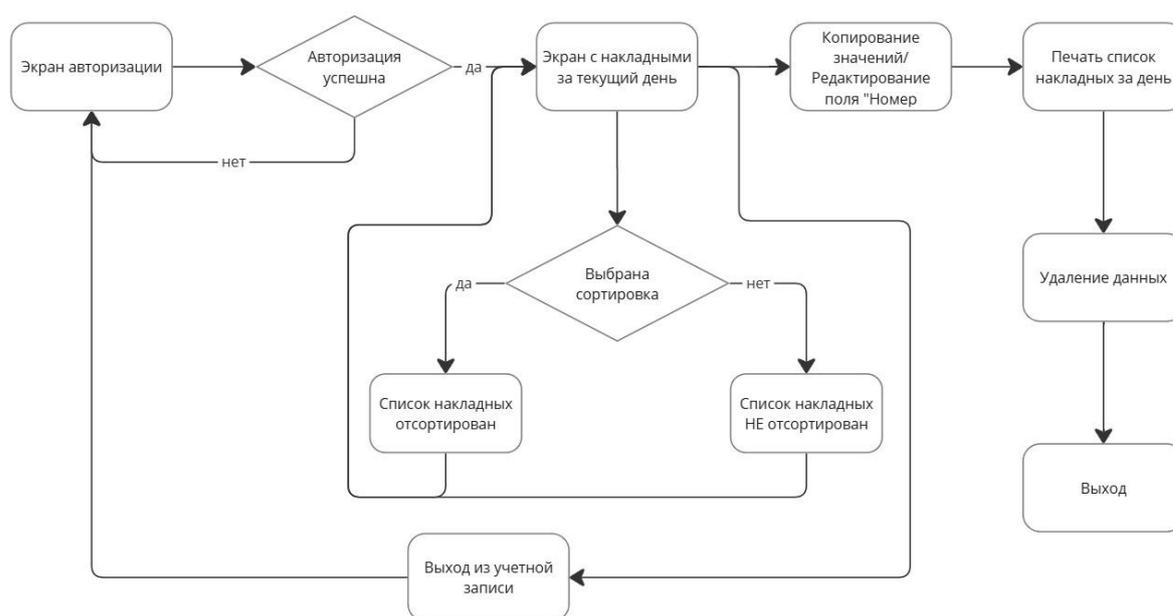


Рисунок 4 – UserFlow ПК-версии

После успешной печати таблицы сотрудник получает возможность удалить текущую базу данных. Это действие необходимо для освобождения системы от обработанных данных и подготовки к следующему рабочему дню. Удаление базы данных возможно только после завершения всех этапов работы, включая печать, чтобы исключить риск потери данных.

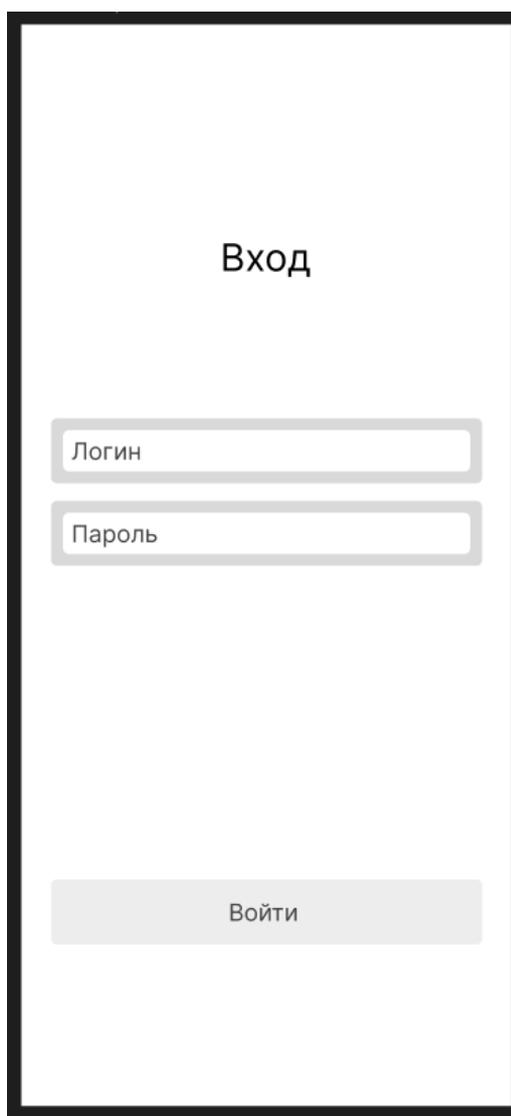
## 4.2 Прототип мобильного приложения

Прототип — быстрая, черновая реализация будущей системы [14]. Прототипы приложений будут реализованы в Figma [15].

Прототип мобильного приложения для системы отслеживания накладных предусмотрены следующие основные экраны:

- экран аутентификации пользователя;
- главный экран;
- экран просмотра и редактирования накладных.

На экране аутентификации (рис. 5) пользователя происходит процесс ввода учетных данных, после чего система проверяет их валидность и предоставляет доступ к функционалу, включая накладные, которые отображаются в приложении. Этот процесс обеспечивает необходимую степень защиты данных и разделение прав доступа между пользователями.



Вход

Логин

Пароль

Войти

Рисунок 5 – Экран аутентификации

Главный экран (рис. 6) приложения является центральным элементом интерфейса и служит для отображения списка накладных, которые автоматически обновляются в течение рабочего дня. Процесс синхронизации происходит в реальном времени, как только бухгалтерия фиксирует заказ в системе.



Рисунок 6 – Главный экран

На верхней панели главного экрана (рис. 6) расположены дополнительные функциональные элементы: поле для поиска, кнопка синхронизации данных и кнопка выхода из учетной записи. Поле поиска позволяет быстро найти необходимую накладную по номеру или дате, что особенно удобно при большом количестве записей. Кнопка синхронизации данных предназначена для ручного обновления информации. Ее использование требуется, когда сотрудник вносит изменения в данные накладной, такие как добавление сведений о выполнении или отгрузке определенного количества товара заказа. После внесения новых данных сотрудник должен нажать кнопку, чтобы обновленная информация была зафиксирована и доступна другим пользователям системы.

Каждая накладная на главном экране представлена в виде карточки или строки с ключевыми полями. В состав отображаемых данных входят: номер накладной, дата, поле с галочкой. Рассмотрим подробнее:

- номер накладной — уникальный идентификатор документа, позволяющий легко его отличить от других записей;
- дата — информация о том, когда был создан заказ;
- поле с галочкой — индикатор, отображающий статус о полном выполнении заказа.

Галочка в поле накладной появляется автоматически, когда пользователь в экране деталей накладной отмечает чек-бокс «Выполнено». Это действие подтверждает, что заказ успешно обработан и завершен. Данный подход позволяет визуальным образом выделить выполненные накладные и облегчает навигацию по списку.

Экран деталей накладной (рис. 7, 8) представляет собой функциональный раздел приложения, который позволяет пользователю получить полную картину о конкретном заказе и при необходимости внести изменения. Этот экран спроектирован для того, чтобы обеспечить максимально удобное и точное управление данными, связанными с заказом.

В верхней части экрана расположены ключевые данные накладной. Здесь крупным шрифтом дублируется номер накладной, который является уникальным идентификатором заказа. Это позволяет пользователю всегда быть уверенным, что он работает с нужным документом. Под номером накладной расположены четыре основных поля, которые содержат основную информацию: «Поставщик», «Покупатель», «Склад», «Дата». Рассмотрим подробнее:

- «Поставщик» — наименование организации, от которой поступил товар;
- «Покупатель» — наименование клиента или компании, для которой предназначен заказ;
- «Склад» — место хранения товара, откуда происходит отгрузка;
- «Дата» — время создания накладной или заказа, что помогает отслеживать сроки и историю операций.

Ниже основной информации размещается таблица, в которой подробно описывается состав заказа. Каждая позиция товара представлена в формате, который включает также 4 поля: «Категория», «Тип», «Кол-во», «Кол-во (уп.)».

- «Категория» — группа, к которой относится товар. К таким категориям относятся: арматура, уголок, швеллер, листы, балки;

- «Тип» — конкретная характеристика товара. Например, арматура может делиться на разные типы в зависимости от диаметра, на 12, 16, 18 и т.д.;
- «Кол-во» — общее количество товара в тоннах;
- «Кол-во (уп.)» — количество товара, измеренное в упаковках.

Таблица позволяет быстро оценить структуру заказа, его объем и состав. Визуальная организация данных способствует их легкому восприятию и быстрому поиску необходимой информации.

← Расходная накладная  
№173210

Поставщик:	“ЕВРАЗ”
Покупатель:	“МАРКА СТАЛИ”
Склад:	Рыбацкая 56
Дата:	19.12.2024

Товар:	
Категория:	Арматура
Тип:	12 (2ф) А500С
Кол-во:	5 т
Кол-во (уп.):	5 т

Отгружено:	
Кол-во:	<input type="text" value="Введите число"/>
Кол-во (уп.):	<input type="text"/>

Выполнено:

Рисунок 7 – Детали накладной

Под таблицей находится интерактивная форма, предназначенная для редактирования информации о выполнении заказа. Сотрудник может вручную указать, сколько товара было отгружено, заполнив два специально отведенных поля: «Кол-во» и «Кол-во (уп.)». Кол-во — фактическое количество отгруженного товара в тоннах. Кол-во (уп.) — фактическое количество отгруженного товара в упаковках. Удобство

редактирования заключается в том, что все изменения фиксируются в реальном времени и могут быть синхронизированы с основной базой данных.

← Расходная накладная  
№173522

Поставщик: "ЕВРАЗ"

Покупатель: "МАРКА СТАЛИ"

Склад: Рыбацкая 56

Дата: 19.12.2024

Товар:

Категория: Арматура

Тип: 12 (2ф) A500C

Кол-во: 5 т

Кол-во (уп.): 5 т

Отгружено:

Кол-во:

Кол-во (уп.)

Товар:

Категория: Арматура

Рисунок 8 – Альтернативный вариант экрана

В нижней части экрана (рис. 7) находится поле с чек-боксом «Выполнено». Сотрудник может поставить галочку в этом поле, чтобы отметить, что заказ полностью выполнен и отгружен. После установки отметки статус накладной изменяется, и она становится помеченной как завершенная в общем списке.

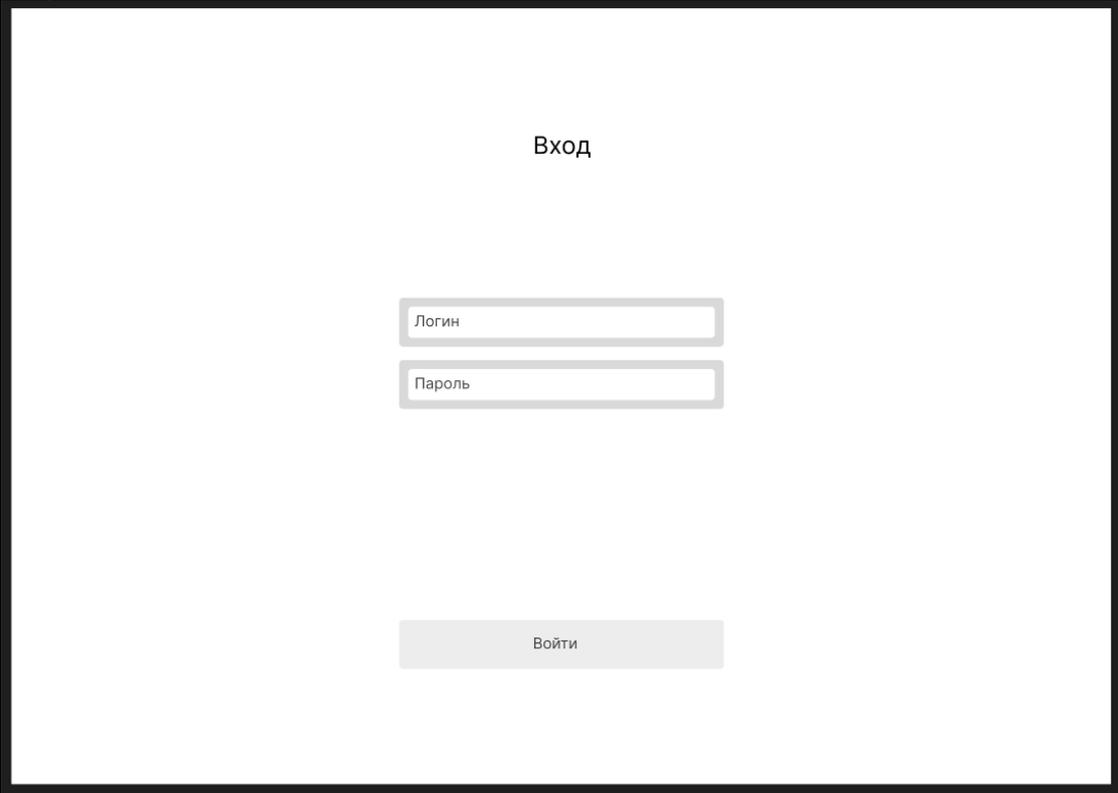
### 4.3 Прототип ПК-версии

Макет программы для ПК-версии спроектирован, чтобы максимально упростить взаимодействие пользователя с системой и обеспечить удобство работы с данными.

Интерфейс состоит из двух главных экранов:

- экран аутентификации пользователя;
- экран со списком накладных.

Первый экран, с которым сталкивается пользователь, — это экран аутентификации (рис. 9). Он служит для ввода учетных данных сотрудника, обеспечивая контроль доступа и защиту данных.



The image shows a simple login interface. At the top center, the word "Вход" is displayed. Below it, there are two rectangular input fields. The first field is labeled "Логин" and the second is labeled "Пароль". At the bottom center, there is a button labeled "Войти". The entire interface is enclosed in a black rectangular border.

Рисунок 9 – Экран аутентификации

Интерфейс экрана на представленном рисунке (рис. 9) максимально прост: поля для ввода логина и пароля, кнопка «Войти». После успешного ввода данных пользователь переходит к основному рабочему экрану.

Основной рабочий экран (рис. 10) состоит из таблицы, которая занимает большую часть пространства. Это позволяет сотруднику сосредоточиться на ключевых данных.

Отгрузка		Склад: ул. Рыбацкая, 56			
Сортировка по:		Дата: 19.12.2024			
№ Документа	№ МХ-3	Наименование			№ Расхода
151607	230045	Арматура 12 (2ф) / "МАРКА СТАЛИ"	14	4960 т	601
151607	230045	Арматура 12 (2ф) / "МАРКА СТАЛИ"	14	4960 т	601
151607	230045	Арматура 12 (2ф) / "МАРКА СТАЛИ"	14	4960 т	601
151607	230045	Арматура 12 (2ф) / "МАРКА СТАЛИ"	14	4960 т	601
151607	230045	Арматура 12 (2ф) / "МАРКА СТАЛИ"	14	4960 т	601
151607	230045	Арматура 12 (2ф) / "МАРКА СТАЛИ"	14	4960 т	601
151596	230039	Арматура 12 (2ф) / "СТРОЙ СНАБ"	14	4960 т	602
151596	230039	Арматура 12 (2ф) / "СТРОЙ СНАБ"	14	4960 т	602
151596	230039	Арматура 12 (2ф) / "СТРОЙ СНАБ"	14	4960 т	602
151596	230039	Арматура 12 (2ф) / "СТРОЙ СНАБ"	14	4960 т	602
151595	230038	Арматура 12 (2ф) / "СТАЛЬГРАД"	14	4960 т	603
151595	230038	Арматура 12 (2ф) / "СТАЛЬГРАД"	14	4960 т	603
151595	230038	Арматура 12 (2ф) / "СТАЛЬГРАД"	14	4960 т	603
151612	230052	Арматура 12 (2ф) / "ТЕКТУМ"	14	4960 т	604

Рисунок 10 – Главный экран

Таблица, отображенная на рисунке 10, организована в несколько колонок:

- «№ Документа»: отображает уникальный номер накладной, что позволяет быстро идентифицировать нужный документ;
- «№ МХ-3»: номер связанного документа, который используется для внутренних процессов. Он выдает автоматически бухгалтерией;
- «№ Расхода»: поле, где сотрудник может внести данные после генерации специального кода системой 1С. Эти поля редактируемы, чтобы информация могла быть оперативно добавлена;

- «Наименование»: это наиболее объемная колонка.

Колонка содержит разделение на три дополнительные подколоники:

- категория, тип и имя заказчика — предоставляет информацию о том, к какой категории относится товар, его тип и кто является заказчиком;
- количество проданных упаковок товара — отображает числовое значение, отражающее объем поставленного товара в упаковках;
- количество проданного товара в тоннах — показывает общий вес товаров в тоннах, отгруженных по данной накладной.

В верхнем левом углу экрана отображается склад, на котором работает сотрудник. Это помогает идентифицировать текущий контекст работы, особенно если сотрудник работает на нескольких складах.

Под указанием склада расположены следующие поля:

- сортировка: позволяет пользователю упорядочить данные таблицы. Вся таблица может быть отфильтрована по имени заказчика. Пользователь может быстро найти нужного клиента и сосредоточиться на работе с его данными;
- дата: отображает текущую дату.

Дополнительное меню (рис. 11), находящееся в крайнем левом углу, включает в себя несколько ключевых функций:

- «Сохранить» — сохраняет все внесенные изменения в базе данных, предотвращая потерю информации при сбоях;
- «Распечатать» — выводит таблицу с накладными на печать, что необходимо для архивирования данных или предоставления их в отчет;
- «Выход» — завершает текущую сессию и возвращает пользователя к экрану аутентификации;
- «Удалить базу» — позволяет очистить базу данных после завершения всех операций. Эта функция доступна только после сохранения и печати данных, чтобы избежать случайного удаления информации.

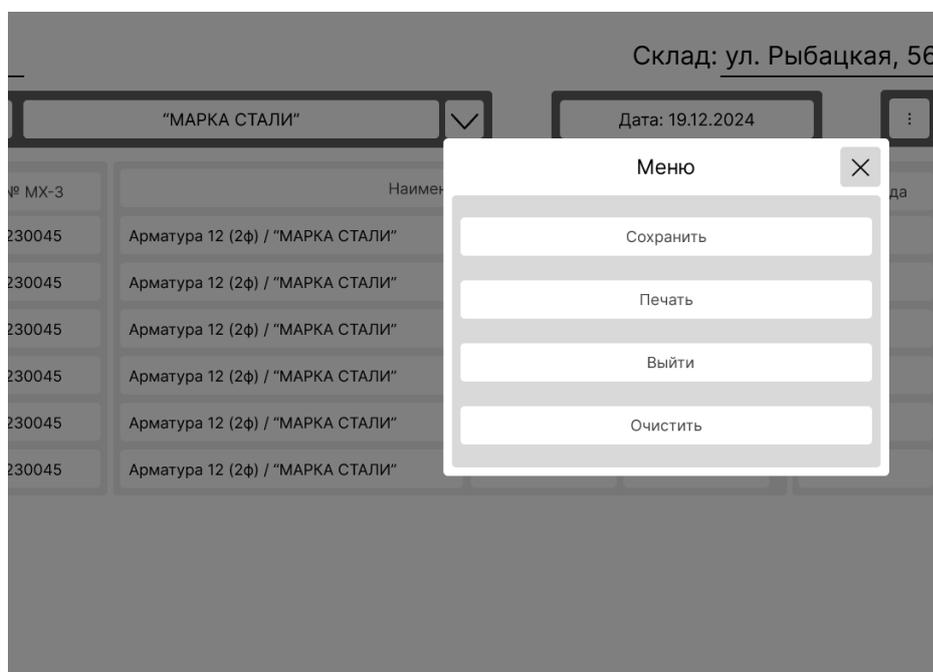


Рисунок 11 – Выпадающее меню

На рисунке 11 представлено выпадающее меню, на котором присутствуют все необходимые функции для работы сотрудников.

## Заключение

В процессе выполнения работы была разработана концепция и прототипы приложений для автоматизации складского учета, включающая мобильное приложение для сотрудников склада и десктопное приложение для обработки и анализа данных. Предложенное решение позволяет сократить работу с бумажными накладными, минимизировать количество ошибок, возникающих при ручной обработке данных, и сократить время выполнения рутинных операций.

Основные результаты работы включают в себя:

- выбор подходящей технологической стека, включая язык программирования и базы данных;
- проектирование интерфейсов приложений.

Тем не менее, в процессе разработки выявлены задачи, требующие дальнейших решений. Одной из таких задач является прямая интеграция приложения с учетной системой 1С, которая играет ключевую роль в управлении данными предприятия. Ограничения, установленные техническим отделом, а также необходимость согласования изменений с руководством в некоторой степени замедляют внедрение этой функции.

Последующие доработки включают в себя:

- адаптация системы к условиям эксплуатации на конкретном складе, включая возможную доработку под необходимые требования;
- учет обратных связей со складом сотрудников для внесения улучшений и корректировок в интерфейсы и функционал;
- разработка и интеграция механической автоматической синхронизации с существующими бухгалтерскими и учетными цепочками, что обеспечивает бесперебойную работу приложений.

Предложенное решение направлено на повышение эффективности работы сотрудников складов. Оно позволяет не только сократить время выполнения стандартных операций, но и создать условия для более точного учета и анализа данных, что в конечном итоге поддерживается общей управляемостью складских процессов.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 Фирма 1С. – Текст: электронный // Фирма 1С: [сайт]. – 2024. – URL: <https://1c.ru/rus/firm1c/firm1c.htm> (дата обращения: 11.12.2024)
- 2 Литовко Ф. Один пример и три нотации: сравниваем BPMN, EPC и DMN. – Текст: электронный / Ф. Литовко // System Education: [сайт]. – 2023. – URL: [https://systems.education/bpmn\\_epc\\_dmn#ercbpmn](https://systems.education/bpmn_epc_dmn#ercbpmn) (дата обращения: 11.12.2024)
- 3 Что такое облачный сервер? – Текст: электронный // Microsoft Azure: [сайт]. – 2022. – URL: <https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-a-cloud-server> (дата обращения: 10.12.2024)
- 4 Википедия [сайт]. – URL: <https://w.wiki/АуКК> (дата обращения: 11.12.2024)
- 5 Кучерявый Е. Что такое API и как он работает. – Текст: электронный / Е. Кучерявый // Skillbox Media: [сайт]. – 2020. – URL: [https://skillbox.ru/media/code/chto\\_takoe\\_api/](https://skillbox.ru/media/code/chto_takoe_api/) (дата обращения: 11.12.2024)
- 6 Протасов К. Что такое диаграмма Ганта и как её построить. – Текст: электронный / К. Протасов, Ю. Слуцкая // Яндекс Практикум: [сайт]. – 2022. – URL: <https://practicum.yandex.ru/blog/diagramma-ganta/> (дата обращения: 12.12.2024)
- 7 Смирнов А. Как создать локальный сервер? Разворачиваем локальный сервер на Windows. – Текст: электронный / А. Смирнов // Nastoysam.Ru: [сайт]. – 2020. – URL: [https://nastoysam.ru/kompyutery/kak-sozdat-lokalnyj-server-razvorachivaem-lokalnyj-server-na-windows?utm\\_source=chatgpt.com#i-5](https://nastoysam.ru/kompyutery/kak-sozdat-lokalnyj-server-razvorachivaem-lokalnyj-server-na-windows?utm_source=chatgpt.com#i-5) (дата обращения: 12.12.2024)
- 8 Denwer [сайт]. – URL: <http://www.denwer.ru> (дата обращения: 03.01.2025)
- 9 Open Server Panel [сайт]. – URL: <https://ospanel.io> (дата обращения: 03.01.2025)
- 10 Википедия [сайт]. – URL: <https://w.wiki/4vGc> (дата обращения: 04.01.2025)
- 11 Википедия [сайт]. – URL: <https://w.wiki/6fMw> (дата обращения: 04.01.2025)
- 12 Мойся Е. Что такое User Flow и как его создать. – Текст: электронный / Е. Мойся, М. Вихрева, П. Овчинникова // Яндекс Практикум: [сайт]. – 2023. – URL: <https://practicum.yandex.ru/blog/chto-takoe-user-flow-i-zachem-ego-razrabatyvat/> (дата обращения: 05.01.2025)
- 13 Miro [сайт]. – URL: <https://miro.com> (дата обращения: 05.01.2025)
- 14 Википедия [сайт]. – URL: <https://w.wiki/8zQE> (дата обращения: 05.01.2025)
- 15 Figma [сайт]. – URL: <https://www.figma.com> (дата обращения: 05.01.2025)

## Приложение А

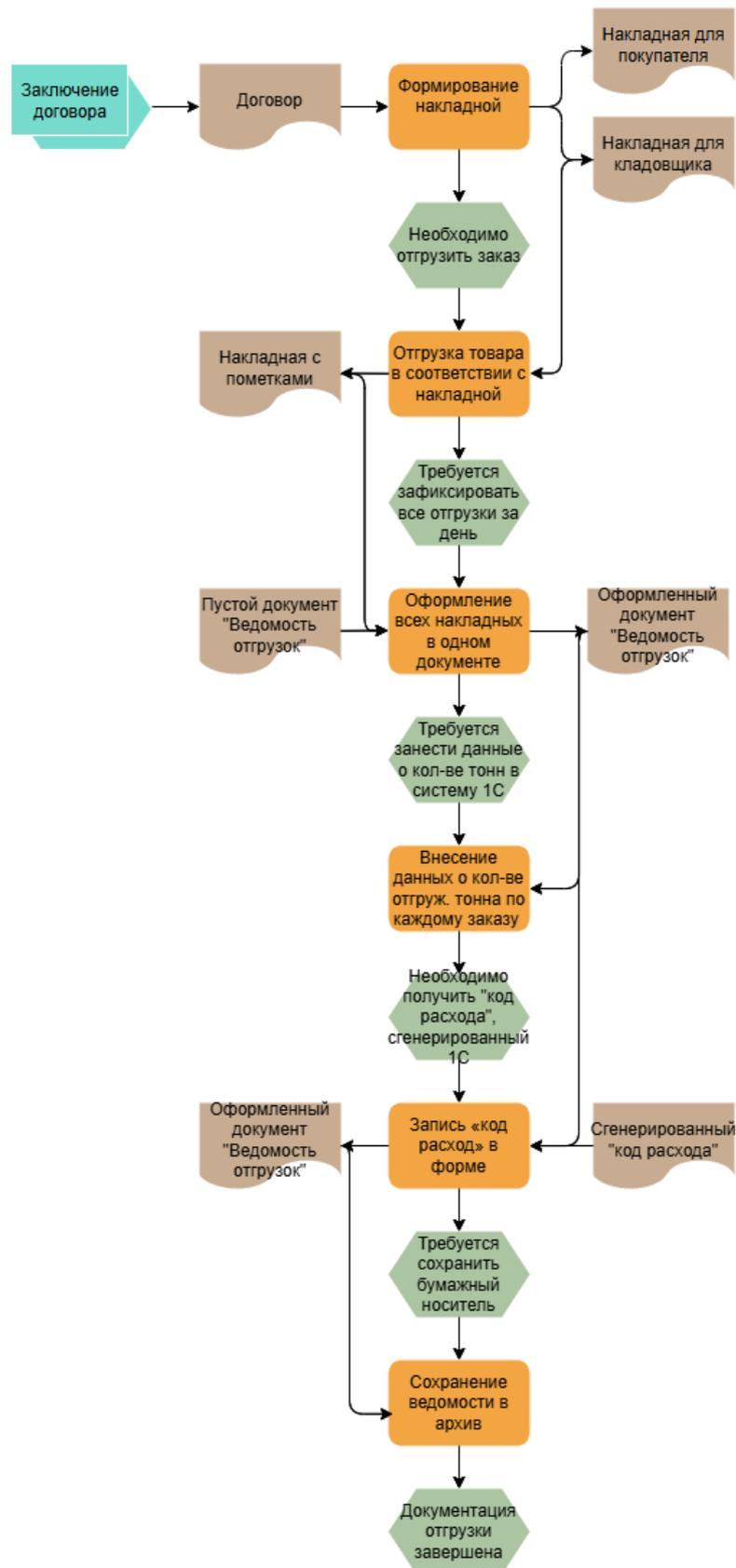


Рисунок А.1 – Представление процесса до автоматизации в нотации EPC

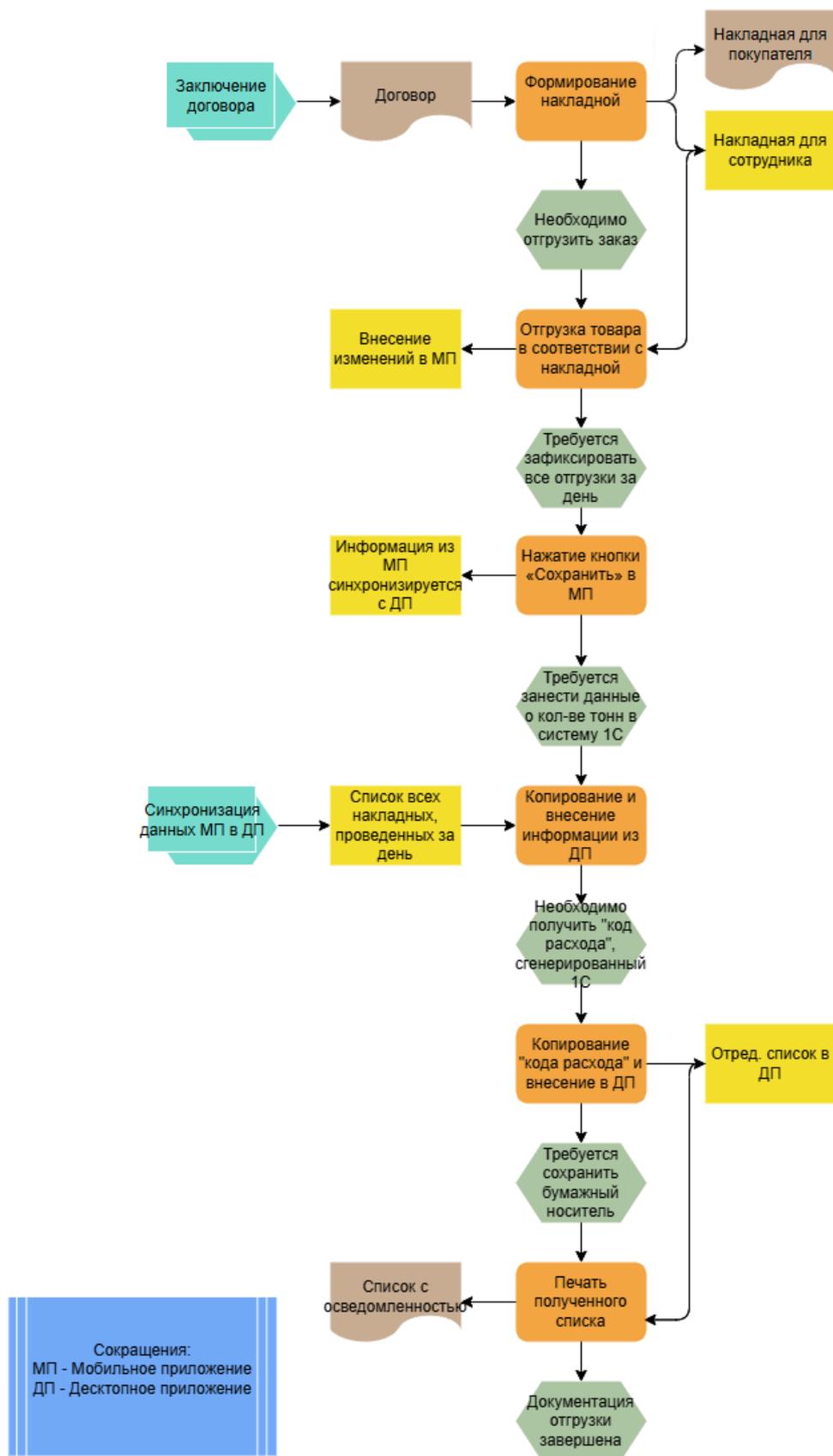


Рисунок А.2 – Представление процесса после автоматизации в нотации EPC