

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка мобильного приложения учета и контроля товара для склада гипермаркета»

Обучающийся

К.А. Щерба

(Инициалы Фамилия)



(личная подпись)

Руководитель

А.П. Тонких

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Тема выпускной квалификационной работы: «Разработка мобильного приложения учета и контроля для склада гипермаркета».

Актуальность работы заключается в необходимости разработки мобильного приложения для учета и контроля на складе гипермаркета.

Цель – разработать мобильное приложение для учета и контроля товара склада гипермаркета.

Для достижения цели необходимо решить следующие задачи:

- осуществить функциональное моделирование процессов учета и контроля склада гипермаркета;
- провести логическое и физическое проектирование мобильного приложения;
- оценить и обосновать экономическую эффективность.

Бакалаврская работа состоит из введения, трех глав, заключения, списка используемой литературы и используемых источников.

Во введении затронуты актуальность работы, методы, объект и предмет исследования, цель работы и задачи для достижения этой цели.

Глава 1 посвящена анализу деятельности склада гипермаркета ООО «Альфа Фуд», поиску основных проблем, бизнес-процессам учета и контроля, обоснованию необходимости разработки автоматизированного варианта решения, формированию требований к новой технологии, анализу конкурирующих приложений и постановке задачи на разработку проекта мобильного приложения.

Глава 2 настоящей работы содержит логическое моделирование мобильного приложения, рассматриваются вопросы информационного обеспечения, проектируется БД и прорабатываются требования к аппаратно-программному обеспечению.

Глава 3 посвящена непосредственно разработке ПО, выбору архитектуры, технологий, базы данных. Описывается основной функционал и проводится оценка и обоснование экономической эффективности.

В заключении рассматриваются цель и задачи работы, описываются результаты исследования.

По итогу проделанной работы реализовано мобильное приложение, при помощи которого:

- автоматизируются процессы учета и контроля на складе гипермаркета;
- уменьшается время выполнения заказов;
- обеспечивается валидация и корректность вводимых данных.

Приложение может быть использовано на складе гипермаркета.

Бакалаврская работа состоит из 63 страниц и включает 38 рисунков, 6 таблиц, 24 источника.

Оглавление

Введение.....	6
Глава 1 Функциональное моделирование учета и контроля товаров на складе гипермаркета.....	8
1.1 Техничко-экономическая характеристика учета и контроля товаров на складе гипермаркета	8
1.2 Концептуальное моделирование учета и контроля товаров на складе гипермаркета	10
1.2.1 Выбор технологии концептуального моделирования предметной области.....	10
1.2.2 Моделирование бизнес-процессов учета и контроля товаров на складе гипермаркета для постановки задачи автоматизированного варианта решения	11
1.2.3 Разработка и анализ модели бизнес-процесса «как есть»	11
1.2.4 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии	14
1.3 Анализ существующих разработок на предмет соответствующих требований	16
1.4 Постановка задачи на разработку проекта мобильного приложения.....	18
1.5 Разработка модели бизнес-процесса «как должно быть».....	19
Глава 2 Логическое проектирование мобильного приложения для учета и контроля товаров склада гипермаркета	22
2.1 Выбор технологии логического моделирования мобильного приложения.....	22
2.2 Логическая модель мобильного приложения и ее описание.....	23
2.3 Информационное обеспечение мобильного приложения	26
2.4 Проектирование базы данных для мобильного приложения	28
2.5 Требования к аппаратно-программному обеспечению.....	30

Глава 3 Физическое проектирование мобильного приложения для учета и контроля на складе гипермаркета.....	32
3.1 Выбор архитектуры мобильного приложения.....	32
3.2 Выбор технологии разработки программного обеспечения	33
3.3 Выбор системы управления базами данных	33
3.4 Разработка физической модели мобильного приложения.....	34
3.5 Разработка программного обеспечения мобильного приложения	36
3.6 Описание функциональности мобильного приложения.....	43
3.7 Оценка и обоснование экономической эффективности разработки мобильного приложения	54
Заключение	58
Список используемой литературы и используемых источников.....	60

Введение

Современное общество сложно представить без использования мобильных телефонов с установленными приложениями. Их популярность обусловлена удобством использования и скоростью получения необходимой информации. Крупные сетевые гипермаркеты часто используют приложения для автоматизации внутренних бизнес-процессов, в том числе и на складах, что позволяет добиться конкурентных преимуществ и повысить прибыльность компании. В разработке мобильного приложения заинтересовано ООО «Альфа Фуд». Компания хочет использовать мобильные устройства для учета и контроля товаров на складе гипермаркета для повышения скорости и качества работы персонала, автоматизации бизнес-процессов.

Актуальность работы обусловлена необходимостью разработки такого мобильного приложения.

Объект исследования – процессы учета и контроля на складе гипермаркета.

Предмет исследования – возможности мобильного приложения для оптимизации учета и контроля товара на складе гипермаркета.

Целью работы является разработка мобильного приложения учета и контроля товара для склада гипермаркета.

Для достижения этой цели необходимо решить спектр следующих задач:

- осуществить функциональное моделирование предметной области;
- произвести физическое и логическое проектирование мобильного приложения;
- привести экономическое обоснование проекта разработки мобильного приложения учета и контроля товара для склада гипермаркета.

Выполнение указанных задач позволит применить разработанное мобильное приложение на складе гипермаркета, оптимизировав тем самым процесс учета и контроля.

Для решения указанных задач применялись методы системного анализа, теории множеств, надежности, вероятностей, оценки рисков, структурно-функционального анализа, теории управления и защиты информации.

Практическая значимость работы заключается в том, что многие торговые организации, имеющие в своем составе склад, испытывают трудности в осуществлении процесса автоматизации его деятельности. Наблюдается необходимость использования мобильного приложения для обеспечения валидации вводимых данных, автоматизации процессов учета и контроля без привязки сотрудников к стационарному рабочему месту.

Структура ВКР обусловлена предметом, целью и задачами исследования, работа состоит из введения, трех глав, заключения, списка используемой литературы и используемых источников.

В первой главе проведен анализ деятельности ООО «Альфа Фуд», разработана и проанализирована модель бизнес-процесса склада гипермаркета «как есть», обоснована необходимость автоматизированного варианта решения, сформированы требования к новой технологии, проанализированы существующие разработки, поставлена задача на разработку мобильного приложения, разработана модель бизнес-процесса «как должно быть».

Во второй главе осуществлено логическое проектирование мобильного приложения, описано информационное обеспечение, спроектирована база данных, заданы требования к аппаратно-программному обеспечению.

Третья глава посвящена физическому проектированию мобильного приложения. Выбрана архитектура, технологии разработки и СУБД. Разработана физическая модель, оценена и обоснована экономическая эффективность разработки мобильного приложения.

В заключении представлены результаты проделанной работы и перспективы использования мобильного приложения.

Итогом ВКР является мобильное приложение для учета и контроля на складе гипермаркета, позволяющее повысить качество и оперативность работы сотрудников склада.

Глава 1 Функциональное моделирование учета и контроля товаров на складе гипермаркета

1.1 Техничко-экономическая характеристика учета и контроля товаров на складе гипермаркета

ООО «Альфа Фуд» – поставщик свежемороженой мясной продукции, рыбной продукции, а также ягод и овощей. Компания занимается крупным и мелким оптом, поставляет продукты в гостиницы, рестораны, кафе, садики и школы. На рынке более 6 лет и имеет хорошую деловую репутацию. Главный склад находится в г. Москва, ул. Рябиновая, 41А.

Площадь складского помещения: 877,6 кв. м.

Среднее ежедневное количество заказов: 84.

Число сотрудников компании: 95 чел.

Компания имеет типичную для торговых предприятий организационную структуру управления, которая представлена на рисунке 1.

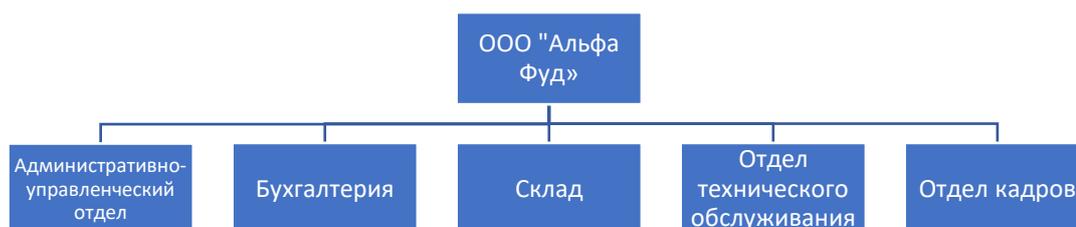


Рисунок 1 – Организационная структура управления ООО «Альфа Фуд»

На складе компании работает 27 человек, из них:

- заведующий склада – 1;
- водители спецтехники – 6 чел. (работают посменно);
- кладовщики – 3 чел.;

- фасовщики – 7 чел.;
- уборщики служебных площадей – 3 чел.;
- грузчики – 7 чел.

Организационная структура склада представлена на рисунке 2. Как видно из этой схемы, все водители, кладовщики, упаковщики, грузчики и уборщики подчиняются непосредственно заведующему складом. Принцип единоначалия в структурном подразделении позволяет контролировать труд каждого специалиста и своевременно корректировать данные ему распоряжения [18].



Рисунок 2 – Организационная структура склада ООО «Альфа Фуд»

Функции, которые выполняют сотрудники склада:

- грузчики перемещают товар по складу, разгружают и загружают фуры;
- фасовщики упаковывают заказы;
- водители спецтехники перемещают паллеты;
- кладовщики занимаются обновлением ассортимента и актуализацией остатков;
- уборщики поддерживают чистоту на складе.

В результате анализа предметной области необходимо разработать мобильное приложение для учета и контроля товара склада данной компании, поскольку этот сектор компании слабо автоматизирован [11]. Реализация

поставленной задачи осуществляется через анализ организации складского учета товаров ООО «Альфа Фуд».

1.2 Концептуальное моделирование учета и контроля товаров на складе гипермаркета

1.2.1 Выбор технологии концептуального моделирования предметной области

Наиболее популярные нотации концептуального моделирования:

- UML,
- IDEF0,
- BPMN.

IDEF0 хорошо подходит для построения процессов верхнего уровня, то есть для показа логического взаимодействия. При этом диаграмма может быть декомпозирована, то есть каждый отдельный элемент представлен как новая схема взаимодействия, детализирующая выбранный блок.

Нотация BPMN показывает движение потоков данных, логику выполнения процессов, изменения определенных величин, процессов и подпроцессов.

Методология UML подходит для создания ПО на любом ООП языке. UML предназначен для визуализации, спецификации, построения и документирования компонентов программных и непрограммных систем. При этом визуализация является наиболее важной ее частью.

В работе удобно использовать нотацию IDEF0, т.к. она обеспечивает описание процессов ООО «Альфа Фуд» и их декомпозицию, а также проста в изучении.

1.2.2 Моделирование бизнес-процессов учета и контроля товаров на складе гипермаркета для постановки задачи автоматизированного варианта решения

На складе гипермаркета ООО «Альфа Фуд» все существующие бизнес-процессы являются результатом обращения менеджеров по продажам.

В результате взаимодействия менеджера с клиентами и поставщиками при помощи программных продуктов «1С: Управление торговлей 8.3» и «1С: Бухгалтерия 8.3» создаются следующие документы:

- приходная накладная,
- расходная накладная,
- реестр товарных позиций,
- отчет об остатках.

Документы распечатываются и передаются на склад.

В фирме четко определены этапы прихода, расхода и контроль остатков.

В целях улучшения качества и скорости учета и контроля предлагается использовать мобильное приложение.

1.2.3 Разработка и анализ модели бизнес-процесса «как есть»

Опишем существующую предметную технологию управления закупками с помощью IDEF0 диаграммы. Контекстная диаграмма приведена на рисунке 3.

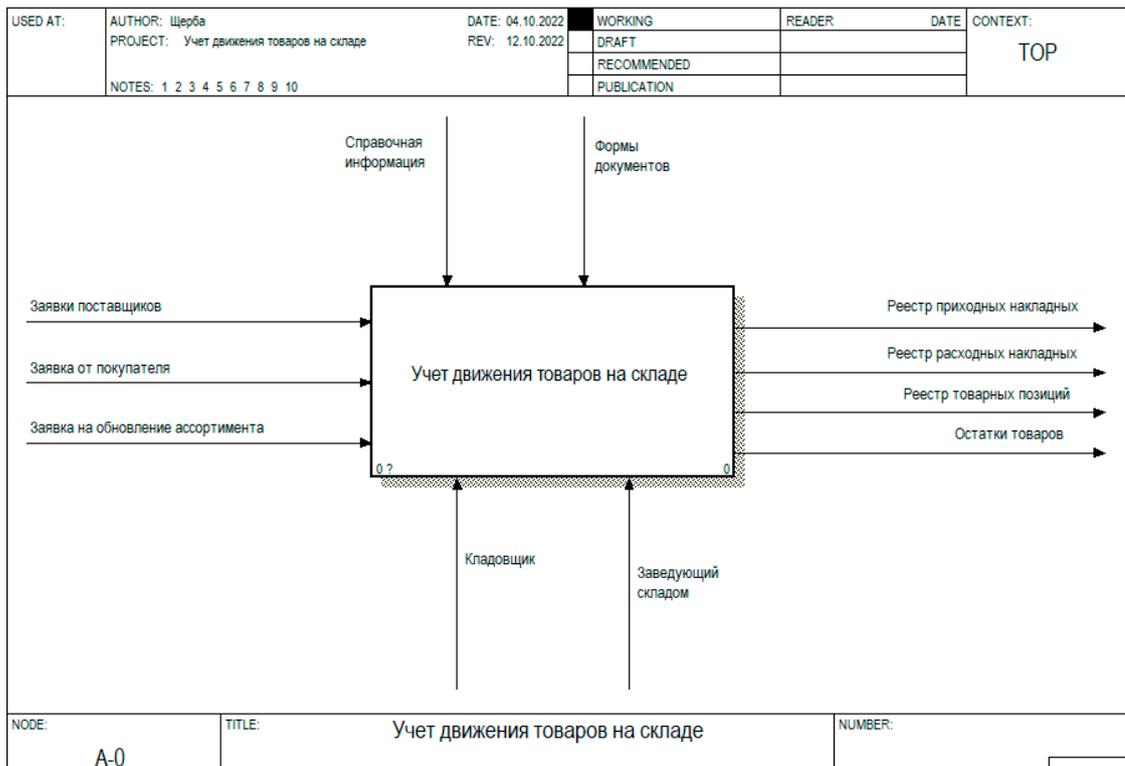


Рисунок 3 – Контекстная диаграмма «Учет движения товаров на складе»

IDEF0-блоки, называемые функциями, обрабатывают и преобразуют входные данные в выходные. IDEF0 моделируют системы как иерархически упорядоченную декомпозицию функций. Диаграмма декомпозиции приведена на рисунке 4.

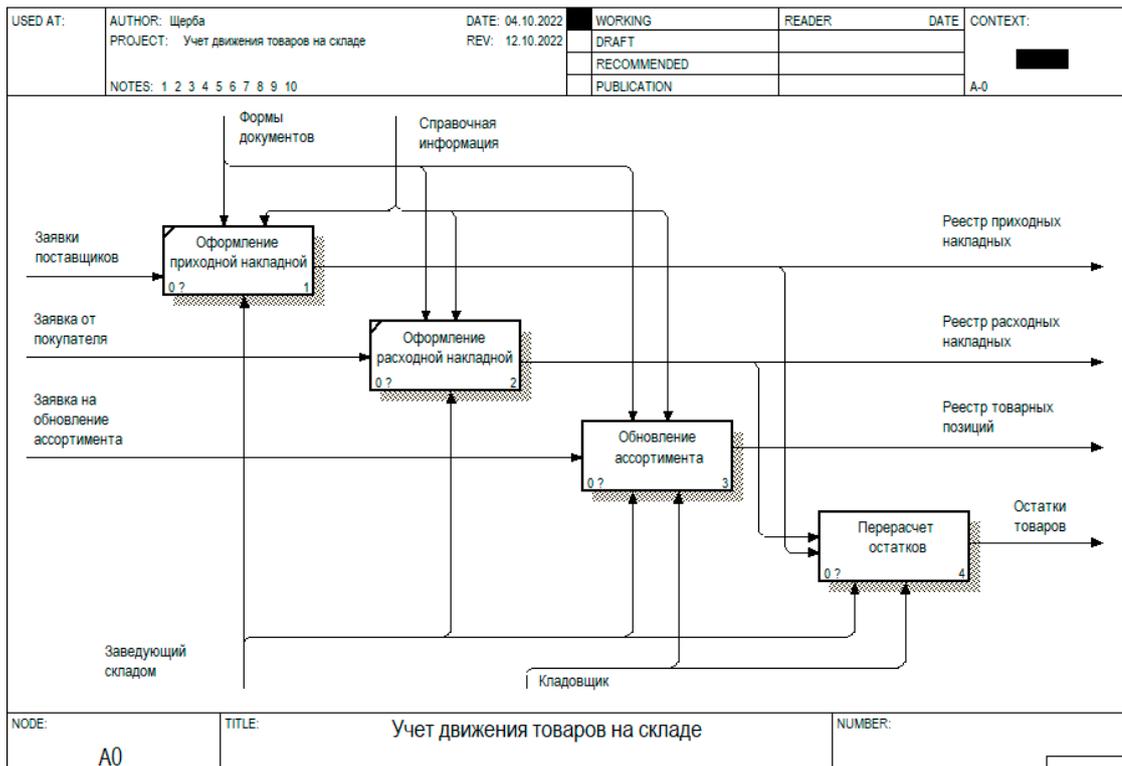


Рисунок 4 – Декомпозиция работы «Учет движения товаров на складе»

С использованием декомпозиции опишем процесс перерасчета остатков товара. Результат отображен на рисунке 5:

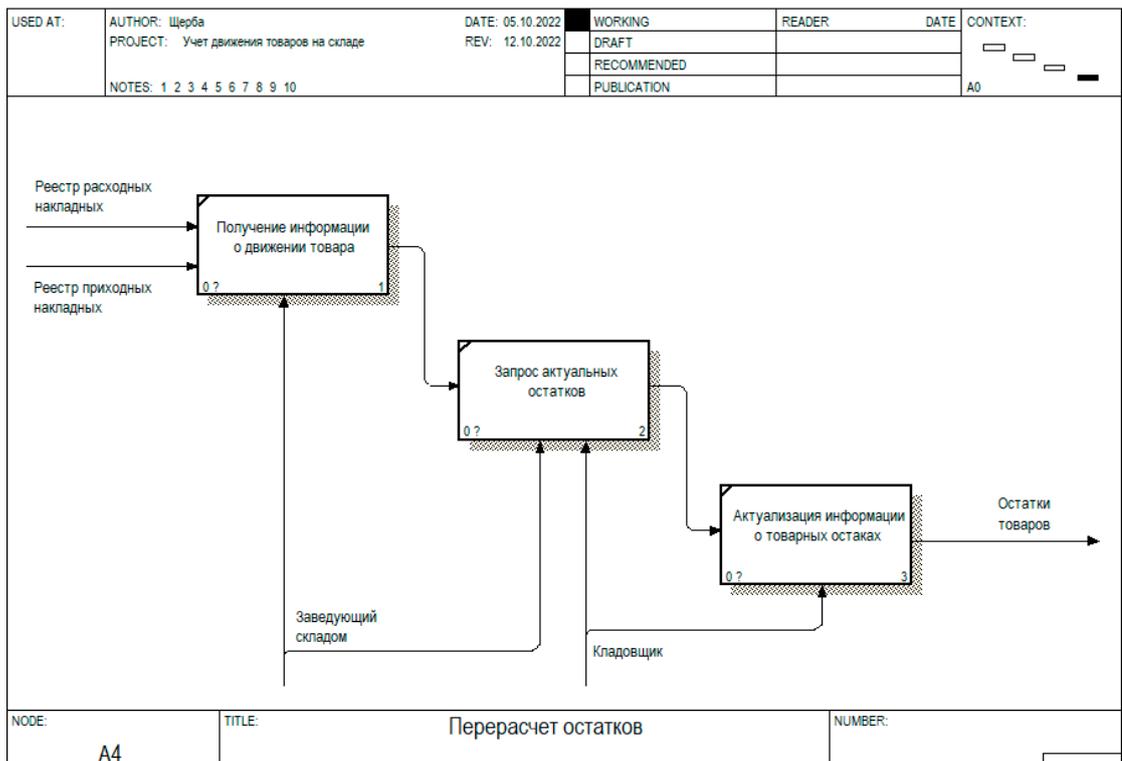


Рисунок 5 – Процесс перерасчета остатков товара AS-IS

Анализируя аспекты складского учета посредством существующего программного продукта (1С: Управление торговлей 8.3), удалось выявить несколько недостатков, создававших неудобство и снижавших скорость обработки информации.

Во-первых, это отсутствие удобного доступа к информации из базы данных. Например, для получения информации об остатках, необходимо делать запрос и получать информацию по электронной почте или в рабочем чате.

Во-вторых, сложный поисковик для поиска товара. Товар приходится искать в полном списке ассортимента склада.

В-третьих, повторное внесение данных, что приводит к возникновению дубликатов.

В-четвертых, система планирования сложная и весьма трудоемкая. Получение прогноза по конкретному товару невозможно, таким образом, быстрое реагирование на изменяющуюся ситуацию также невозможно.

В-пятых, в планировании продаж были большие погрешности, поскольку не все факторы могли быть учтены при прогнозировании.

1.2.4 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

Внесем в таблицу 1 временные характеристики процессов, о которых говорили ранее.

Таблица 1 – Временные характеристики процессов

Процессы	Примерное количество в день	Время, затрачиваемое на выполнение, минуты	Итого, минуты
Оформление приходной накладной	5	60	300
Оформление расходной накладной	5	60	300

Продолжение таблицы 1

Процессы	Примерное количество в день	Время, затрачиваемое на выполнение, минуты	Итого, минуты
Обновление ассортимента товаров	5	30	150
Перерасчет товарных остатков	15	30	450
ИТОГО, минут			1200

Таким образом, в среднем, 1200 минут каждый день кладовщики и заведующий складом заняты обработкой заказов. При этом на реализацию других рабочих задач у сотрудников остается не так много времени. В случае высокой загрузки работодатель вынужден привлекать персонал для работы во внеурочное время, что приводит к дополнительному незапланированному расходу из фонда оплаты труда.

В таблице 2 рассматривается эффект от применения мобильного приложения для учета и контроля товаров на складе.

Таблица 2 – Определение эффекта от внедрения мобильного приложения

Процессы	Примерное количество в день	Время, затрачиваемое на выполнение, минуты	Итого, минуты
Оформление приходной накладной	5	30	150
Оформление расходной накладной	5	30	150
Обновление ассортимента товаров	5	15	75
Перерасчет товарных остатков	15	15	225
ИТОГО, минут			600

Экономия времени может составить до 600 минут.

Другие преимущества:

- экономия времени по перемещению от стационарного рабочего места и обратно;
- система авторизации и разграничения доступа;
- верификация вводимых данных [1], [8].

Для усовершенствования деятельности склада гипермаркета ООО «Альфа Фуд» необходимо разработать мобильное приложение, выполняющее следующие задачи:

- учет и контроль информации о приемке и отгрузке товара;
- ведение ассортимента;
- учет товарных остатков;
- авторизацию пользователей с разграничением прав доступа.

Рассмотрим и проанализируем существующие мобильные приложения.

1.3 Анализ существующих разработок на предмет соответствующих требований

Определим критерии анализа мобильных приложений, которые уже представлены на рынке, по следующим параметрам:

- поддержка клиент-серверной технологии;
- разграничения доступа пользователей;
- возможность указать место товара на складе.

Для анализа предлагаются следующие продукты:

- AdvancePro Inventory,
- Scandit Flow,
- EazyStock.

Программное обеспечение AdvancePro Inventory предназначено для упрощения трудовых обязанностей менеджеров склада с помощью инструментов управления запасами, позволяющих управлять заказами, эффективностью и информацией в режиме реального времени.

Скриншот программы показан на рисунке 6.

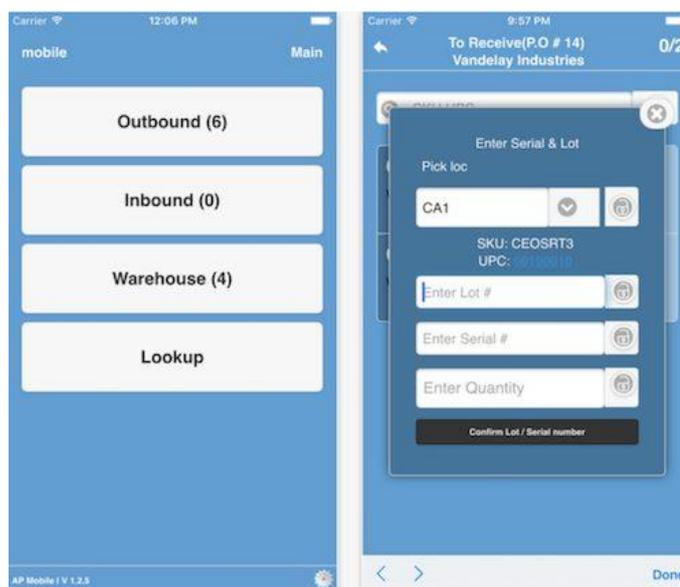


Рисунок 6 – Интерфейс программы AdvancePro Inventory

Scandit Flow – настраиваемое через интернет приложение для управления складом, созданное для поддержки настраиваемых корпоративных рабочих процессов.

Интерфейс программы показан на рисунке 7.

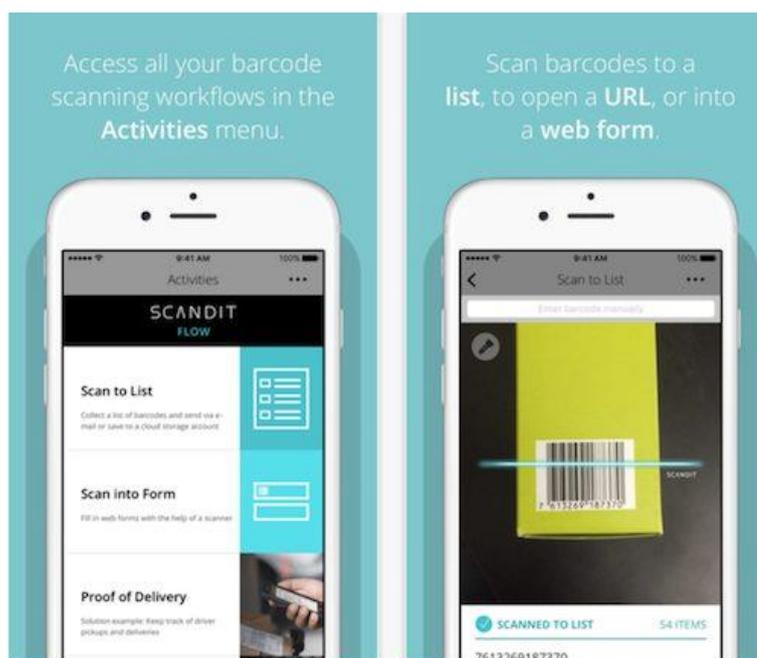


Рисунок 7 – Интерфейс программы Scandit Flow

EazyStock — это облачная система оптимизации складских остатков, которая обеспечивает экономию затрат при одновременном повышении уровня обслуживания клиентов.

Сравнительный анализ по выбранным критериям мобильных приложений показан в таблице 3.

Таблица 3 – Сравнительный анализ существующих мобильных приложений

Наименование	Поддержка клиент-серверной технологии	Разграничения доступа пользователей	Возможность указать место товара на складе
AdvancePro Inventory	-	-	+
Scandit Flow	+	+	-
EazyStock	+	-	+

По результатам анализа существующих мобильных приложений можно сделать вывод, что не существует программы, полностью удовлетворяющей требованиям. То есть для учета и контроля товаров на складе гипермаркета ООО «Альфа Фуд» необходимо разработать приложение, которое даст возможность оптимизировать работу персонала.

1.4 Постановка задачи на разработку проекта мобильного приложения

Рассмотрим постановку задачи автоматизации учета и контроля товаров на складе гипермаркета ООО «Альфа Фуд» [4].

Мобильное приложение должно автоматизировать процесс работы склада и решать следующие задачи:

- сокращение временных затрат по приемке и отгрузке товара;
- учет товарных остатков;
- сокращение использования бумажных носителей.

Мобильное приложение должно обеспечить отправку и выдачу данных по запросам сотрудников:

- предоставить возможность ввода информации, проверки ее на корректность, а также обеспечить вывод;
- осуществлять запросы по критериям сотрудников склада;
- обеспечивать аутентификацию пользователей и разделение прав.

Цель внедрения приложения:

- верифицировать входную информацию для уменьшения количества ошибок;
- быстро обрабатывать поступающие запросы для экономии времени;
- сократить фонд оплаты труда за счет автоматизации.

Функции системы:

- автоматизация приходных и расходных операций;
- ведение ассортимента товаров и их остатков;
- обеспечение контроля доступа к данным.

Для реализации поставленных задач перейдем к разработке модели бизнес-процесса «ТО ВЕ».

1.5 Разработка модели бизнес-процесса «как должно быть»

Для автоматизации деятельности склада гипермаркета применим новый механизм – мобильное приложение.

Нарисуем функциональные диаграммы для понимания того, как изменится процесс работы склада гипермаркета.

В результате внедрения мобильного приложения получим автоматизацию основного процесса, как показано на рисунке 8.

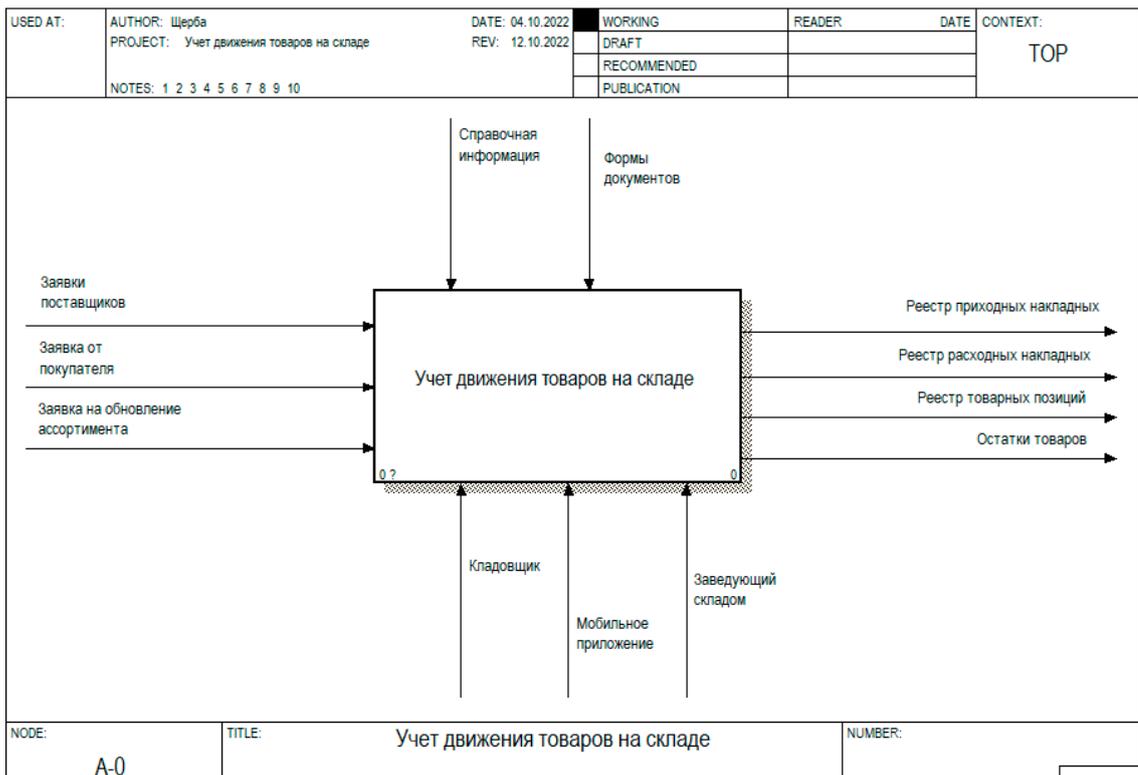


Рисунок 8 – Контекстная диаграмма ТО-ВЕ

На рисунке 9 показана декомпозиция контекстной диаграммы.

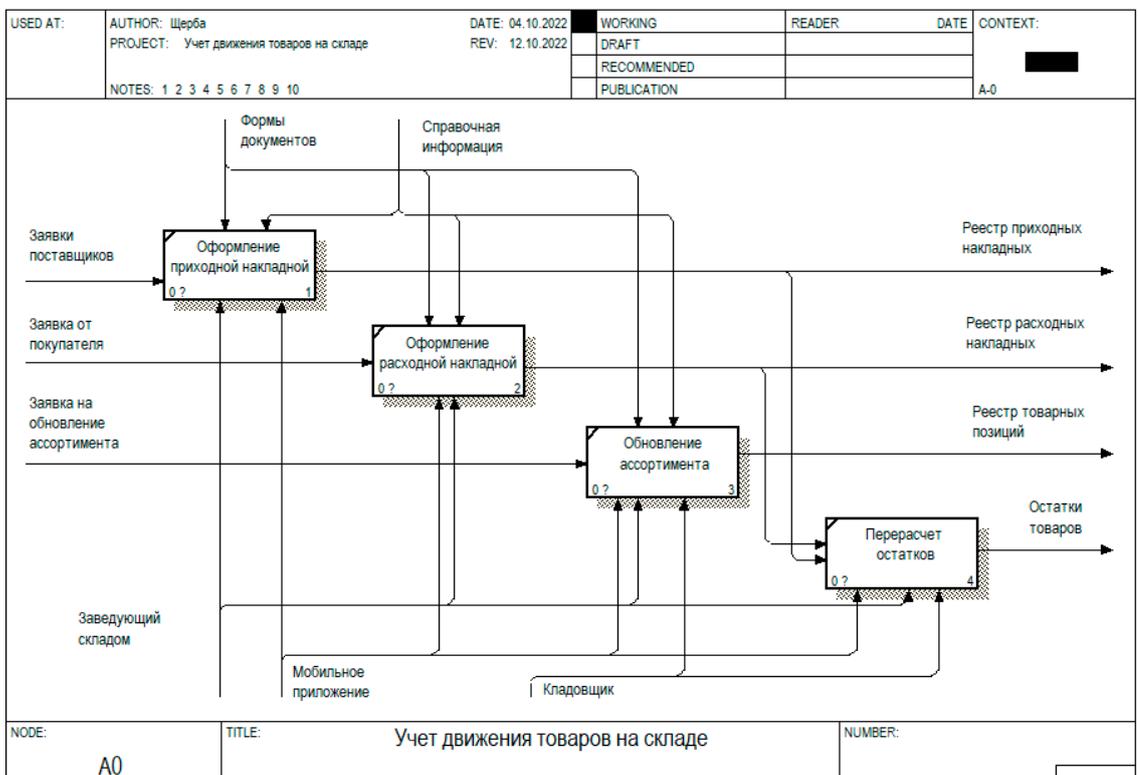


Рисунок 9 – Декомпозиция процесса учета и контроля товаров на складе гипермаркета ТО-ВЕ

На этой диаграмме отображен процесс учета и контроля на складе гипермаркета более подробно.

Выводы по главе 1

В первой главе ВКР рассмотрена деятельность предприятия ООО «Альфа Фуд», дана характеристика его работы, проведен обзор мобильных приложений, которые применяются в складском учете и контроле товаров.

В результате сравнения аналогов мобильных приложений сделан вывод, что для удовлетворения требований по учету и контролю товаров на складе гипермаркета не существует решений на свободном рынке.

Для оптимизации осуществления процедур контроля и учета на складе гипермаркета нужно разработать мобильное приложение, которое даст возможность оптимизировать деятельность предприятия в сфере функционирования персонала склада.

При этом решаются такие задачи, как:

- уменьшение количества ошибок учета и контроля;
- сокращение времени обработки типичных операций;
- уменьшение фонда оплаты труда.

Глава 2 Логическое проектирование мобильного приложения для учета и контроля товаров склада гипермаркета

2.1 Выбор технологии логического моделирования мобильного приложения

Для создания логической модели (с целью снижения количества системных ошибок, сокращения общего объема и времени разработки) применяются CASE-средства.

Преимущества использования CASE-средств:

- автоматизация этапов проектирования;
- повышение качества программного кода и самого проекта;
- согласованность и полнота документации;
- легкий перенос наработок в другие проекты [15].

Для создания логической модели мобильного приложения хорошо подойдет CASE-средство StarUML. Это инструмент разработки программного обеспечения с использованием языка UML. В отличие от подобных программ, таких как Rational Rose или ERwin, продукт бесплатен и имеет простой пользовательский интерфейс, который понятен даже неподготовленному пользователю. При необходимости возможности StarUML можно расширить при помощи системы плагинов.

UML позволяет создавать графические модели, основываясь на принципах объектно-ориентированного подхода, в отличие от IDEF0, где применяется структурный подход.

Так как в процессе логического проектирования планируется применить объектно-ориентированный подход, остановим свой выбор на UML и CASE-средстве StarUML, что позволит создать архитектуру всех процессов учета и контроля на складе предприятия учитывая взаимосвязи и управляющие

воздействия между уровнями иерархии.

2.2 Логическая модель мобильного приложения и ее описание

Для создания логической модели ИС нужно создать ряд основополагающих диаграмм.

Первая диаграмма, необходимая для создания логической модели – диаграмма вариантов использования. Она позволяет описать систему на концептуальном уровне, отражает отношения между ролями и прецедентами.

Диаграмма вариантов использования отражена на рисунке 10.

Роли, показанные на диаграмме:

- кладовщик – уполномоченное компанией лицо, ответственное за работу с товарным ассортиментом;
- заведующий складом – сотрудник ответственный за учет приема и расхода товаров со склада, описание его секций, работу с проблемами, а также за предоставление доступа кладовщикам к работе со списком товаров.



Рисунок 10 – Диаграмма вариантов использования

Характеристика прецедентов представлена в таблице 4.

Таблица 4 – Характеристика прецедентов

Прецедент	Характеристика
Работа со списком товаров	Работа с товарным ассортиментом. Добавление, удаление и редактирование списка товаров на складе. Учет их весовых и объемных характеристик.
Отчет о проблеме	Возможность отправки сообщения при нештатных ситуациях на складе.
Приход товара	Фиксация поступления товара на склад.
Расход товара	Фиксация отгрузки товара со склада.
Работа со списком сотрудников	Добавление, удаление и редактирование кладовщиков, присвоение ролей.
Работа со списком секций	Добавление, удаление и редактирование секций склада гипермаркета.
Работа со списком проблем	Обработка проблем, поступивших от кладовщиков.

На рисунках 11, 12 изображены диаграммы последовательности для заведующего складом и кладовщика соответственно. Задача данной диаграммы – визуализировать последовательность действий при работе с мобильным приложением. Заведующий складом может осуществлять приемку и отгрузку товаров, управлять пользователями, секциями и проблемами. Кладовщик управляет ассортиментом товаров и имеет возможность сообщить о проблеме. Пользователи системы должны пройти обязательную авторизацию.

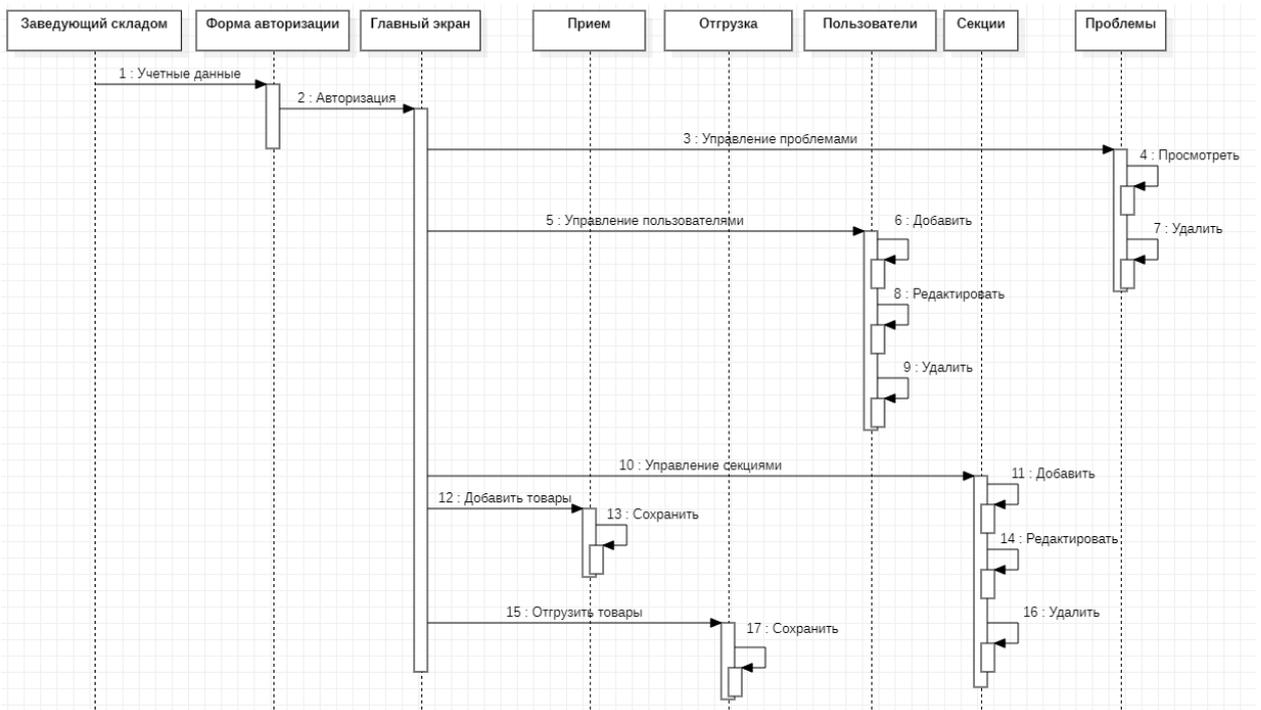


Рисунок 11 – Диаграмма последовательности для заведующего складом

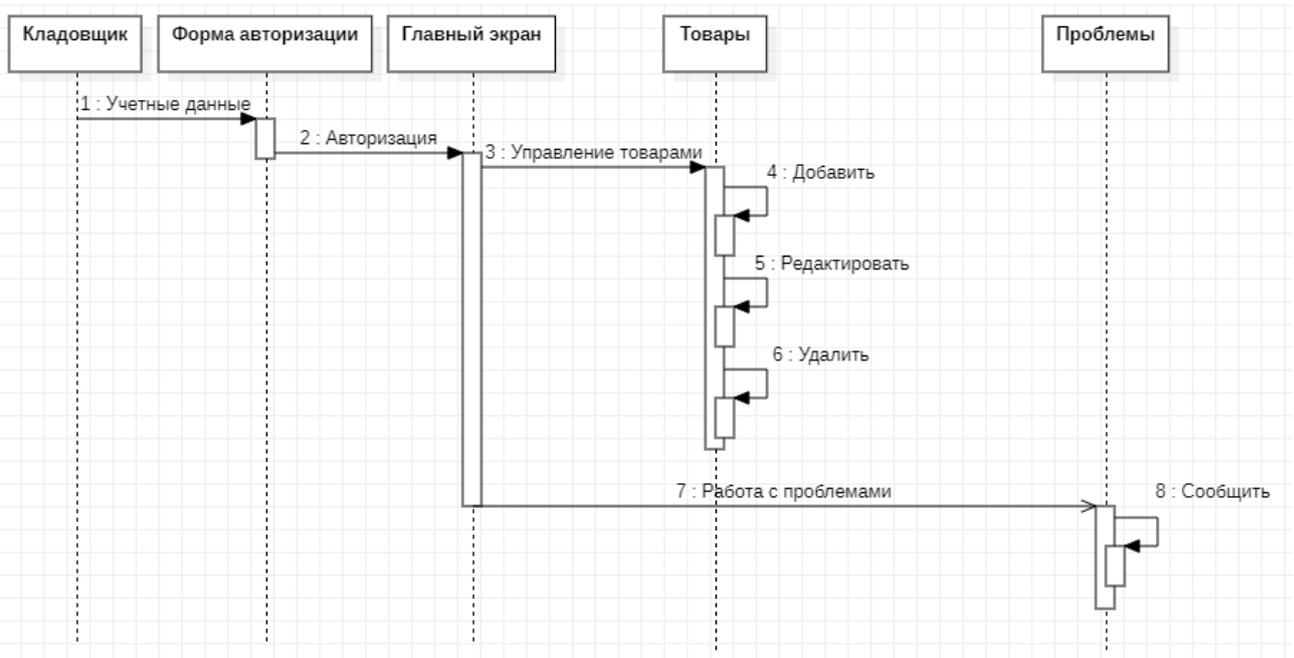


Рисунок 12 – Диаграмма последовательности для кладовщика

На рисунке 13 показана диаграмма классов, на которой отображены атрибуты и операторы классов, а также их взаимосвязь.

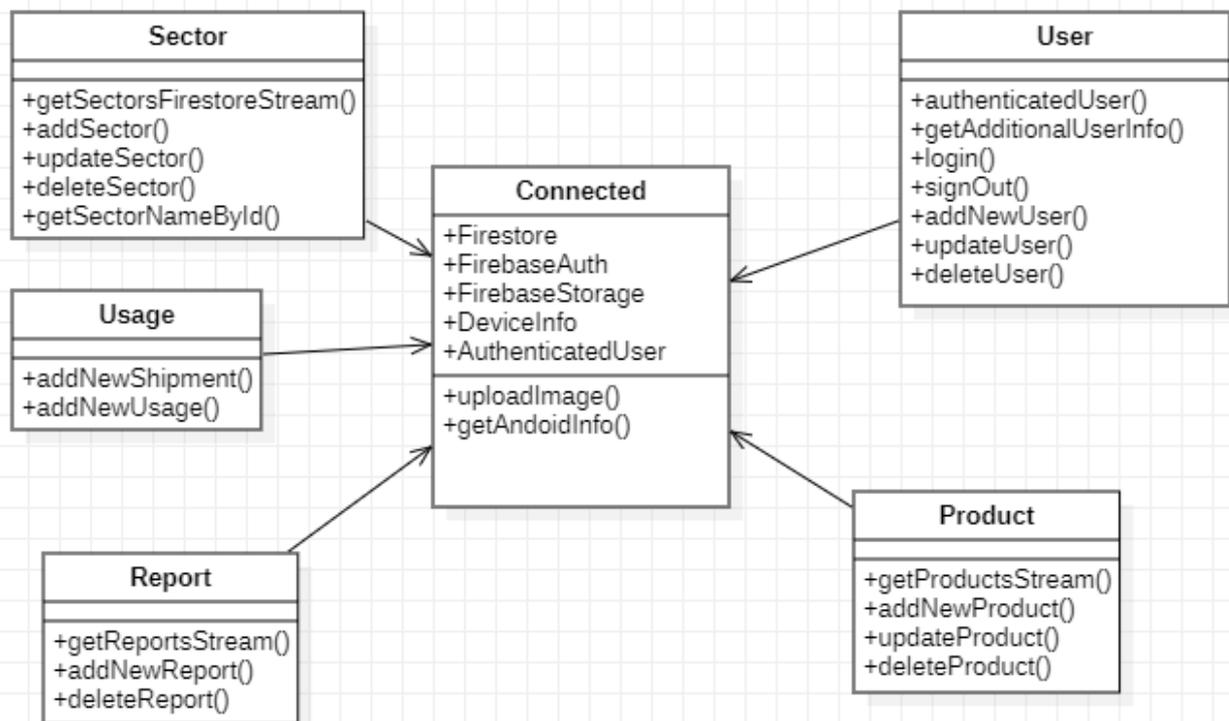


Рисунок 13 – Диаграмма классов

На диаграмме классов изображены такие классы как:

- connected – отвечает за соединение с СУБД и авторизацию пользователей,
- product – реализация методов работы с товаром,
- user – управление пользователями,
- sector – управление секциями,
- report – отчет о проблеме,
- usage – отгрузка и приемка товара.

На основе приведенных диаграмм будет строиться логическая модель мобильного приложения.

2.3 Информационное обеспечение мобильного приложения

Входными документами для реализации мобильного приложения учета и контроля товаров на складе гипермаркета являются:

Все документы для автоматизации процессов учета и контроля, поступающие в ООО «Альфа Фуд» подчиняются согласованной заранее структуре и имеют табличную форму.

2.4 Проектирование базы данных для мобильного приложения

На этапе проектирования базы данных решается задача определения необходимых сущностей и их полей, а также связей между ними.

Сущности:

- workers – список сотрудников организации, таких как заведующий складом и кладовщиков,
- reports – сообщения о проблемах при осуществлении учета и контроля на складе,
- products – ассортимент товаров гипермаркета,
- sectors – описание секций склада,
- receipt – хранение информации о поступлении и выдачи товаров со склада.

Описания полей сущностей базы данных приведены в таблице 5.

Таблица 5 – Описание сущностей

Поле сущности	Описание
Сущность Workers	
id	Код работника
address	Адрес
adminOrUser	Признак администратора
email	Электронная почта
imageUrl	Фотография
name	Имя
phone	Телефон
Сущность Reports	
id	Код проблемы
description	Описание

Продолжение таблицы 5

Поле сущности	Описание
time	Время возникновения
userReported	Код пользователя
Сущность Products	
id	Код продукта
barcode	Штрихкод
description	Описание
imageUrl	Изображение
measurementUnit	Код секции
name	Имя
quantity	Количество
wherelsStored	Единица измерения
Сущность Sectors	
id	Код секции
description	Описание
name	Имя
Сущность Receipt	
id	Код получения/отгрузки
from	Номер накладной
productsArrived	Массив товаров
time	Время
totalPrice	Суммарная стоимость
userCreated	Код пользователя

Отообразим связи между сущностями на рисунке 16 при помощи ER-диаграммы.

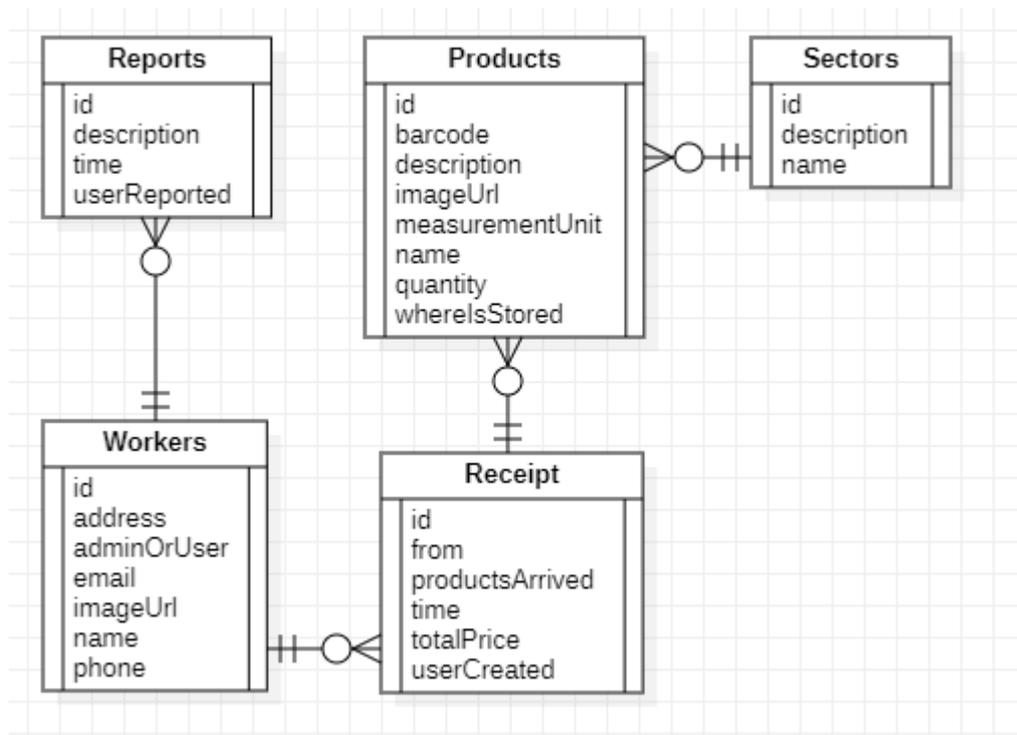


Рисунок 16 – ER-диаграмма

Таким образом получим наглядное представление о сущностях, их полях и связях между ними.

2.5 Требования к аппаратно-программному обеспечению

Укажем минимальные системные требования к аппаратно-программному обеспечению для работы мобильного приложения:

- телефон или планшет на базе Android не менее версии 7.0;
- 1 Гб оперативной памяти;
- 50 Мб свободного пространства на flash-накопителе;
- доступ в сеть интернет для соединения с сервером БД [7].

Для разработки мобильного приложения требуется персональный компьютер со следующими характеристиками:

- операционная система Windows 10,
- процессор Intel Core i5 не ниже 4-ого поколения,
- 16 Гб оперативной памяти,

- 120 Гб на жестком диске,
- высокоскоростное подключение к сети интернет для загрузки необходимых библиотек и средств разработки [10].

Выводы по главе 2

В главе 2 выпускной квалификационной работы было выбрано CASE-средство логического моделирования StarUml, дана характеристика логической модели, спроектирована БД, заданы требования к аппаратно-программному обеспечению.

Разработанные диаграммы являются основой логической модели данных, полученная логическая модель будет использована для построения физической модели деятельности склада гипермаркета.

На основе логической модели данных можно перейти к физическому проектированию мобильного приложения для склада гипермаркета ООО «Альфа Фуд».

Глава 3 Физическое проектирование мобильного приложения для учета и контроля на складе гипермаркета

3.1 Выбор архитектуры мобильного приложения

В мобильной разработке распространены два вида архитектур: клиент-серверная (взаимодействие множества клиентов с одним или несколькими серверами) и без выделенного сервера (все данные хранятся внутри приложения).

Так как мобильное приложение будет установлено на несколько устройств и будет использоваться несколькими сотрудниками склада одновременно, а также необходимо обеспечить доступ к общей базе данных, применим клиент-серверную архитектуру [9].

На рисунке 17 отображена такая архитектура. Информация поступает в двух направлениях, а один сервер может общаться сразу с несколькими клиентами.

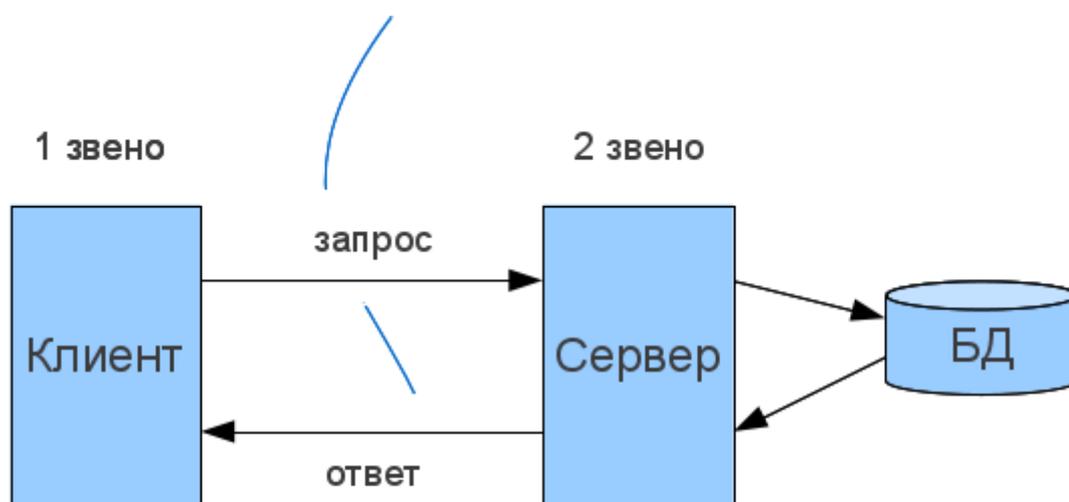


Рисунок 17 – Клиент-серверная архитектура

В высоконагруженных системах отдельные функции могут быть вынесены на специализированные сервера, что способствует более эффективному их использованию.

3.2 Выбор технологии разработки программного обеспечения

Для выбора технологии разработки программного обеспечения проведем сравнительный анализ существующих на рынке фреймворков [12].

Наиболее популярными технологиями разработки мобильных приложений без привязки платформе являются:

- Flutter – написанный на языке Dart,
- React Native – подразумевает использование языка JavaScript.

Разработка гибридных мобильных приложений значительно экономит время и бюджет заказчика.

React Native – популярное средство разработки с открытым исходным кодом. Фреймворк позволяет создавать приложения для Android и iOS [6], [14], [21], [23]. Использует знакомый многим язык JavaScript.

Flutter – поддерживает аналогичные платформы, при этом обеспечивая менее трудоемкое создание множества экранов за счет системы виджетов и более качественный вид приложений на старых устройствах [22].

В связи с тем, что процесс создания экранов в React Native требует больших трудозатрат и времени на разработку, а их количество будет более 10 – остановим выбор на Flutter.

3.3 Выбор системы управления базами данных

Так как предполагается работать с большим количеством документов – хорошим выбором будет использование нереляционной СУБД, такой как Firestore или MongoDB.

Документно-ориентированная база данных в отличие от классической реляционной не содержит таблиц. Данные хранятся в сущностях, называемых документами.

Документ представляет собой набор пар ключ-значение. База хорошо подходит для хранения большого количества маленьких документов.

Отличительной особенностью баз NoSQL является возможность построения иерархических документов, т.е. включение одного документа внутрь другого.

MongoDB доступна в исходном коде и требует установки на сервер.

Firestore – полностью облачное решение, возможностей бесплатного тарифного плана вполне достаточно для хранения данных предприятия ООО «Альфа Фуд». В случае необходимости можно с легкостью приобрести дополнительные ресурсы.

Дополнительное удобство обеспечивает неявное создание коллекций и документов. Достаточно просто отправить данные. Если коллекция или документ не существуют, Firestore создает их.

Остановим свой выбор на Firestore, так как СУБД обеспечивает легкую работу с большим количеством документов, поддерживает иерархическую структуру и не требует при этом покупки отдельного сервера.

3.4 Разработка физической модели мобильного приложения

Выполним разработку физической модели базы данных для мобильного приложения в понятиях Firestore Database и внесем информацию в таблицу 6:

Таблица 6 – Описание полей документа

Поле документа	Тип данных
Сущность Workers	
id	String
address	String
adminOrUser	String
email	String
imageUrl	String
name	String
phone	String
Сущность Reports	
id	String
description	String
time	String
userReported	String
Сущность Products	
id	String
barcode	String
description	String
imageUrl	String
measurementUnit	String
name	String
quantity	Integer
whereIsStored	String
Сущность Sectors	
id	String
description	String
name	String
Сущность Receipt	
id	String
from	String
productsArrived	String
time	String
totalPrice	Integer
userCreated	String

Физическая модель мобильного приложения показана на рисунке 18.

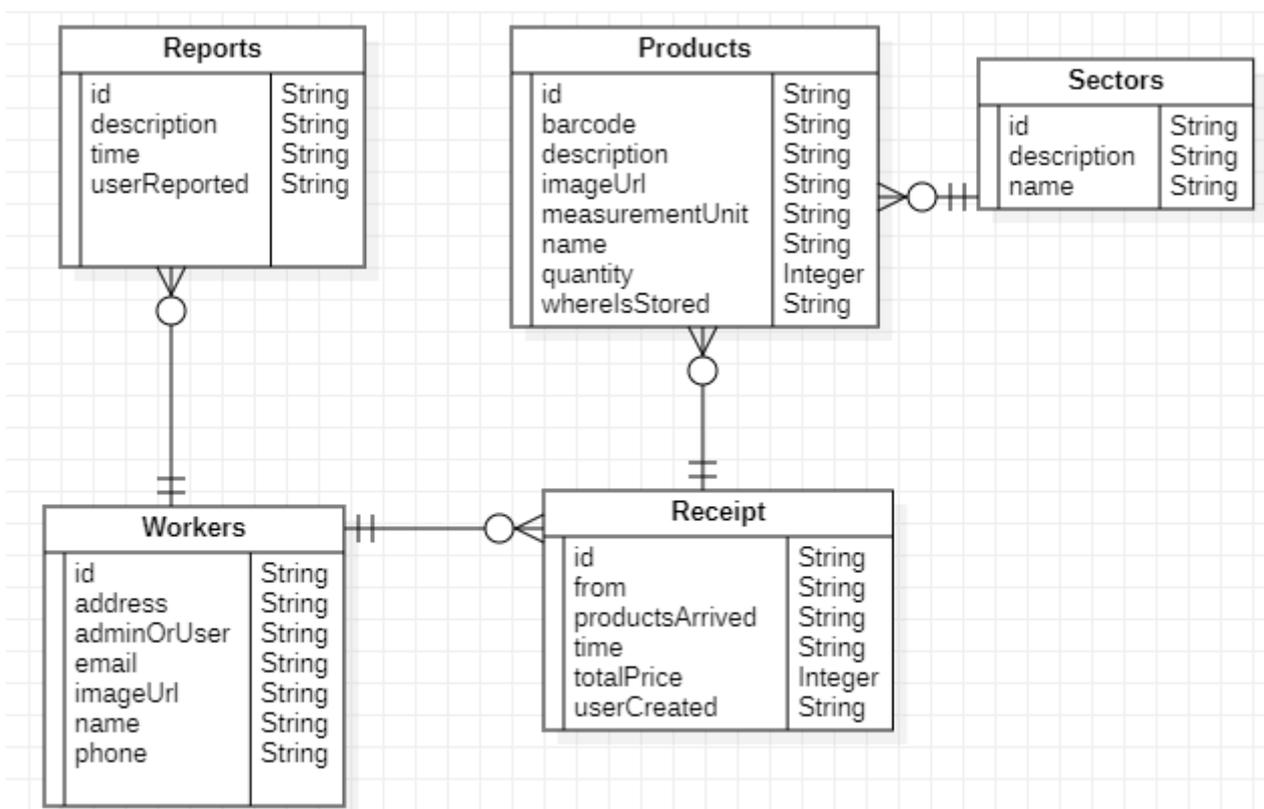


Рисунок 18 – Физическая модель мобильного приложения

На ней показаны сущности базы данных и связи между ними.

3.5 Разработка программного обеспечения мобильного приложения

Для разработки клиентской части мобильного приложения будем использовать фреймворк Flutter, язык программирования Dart и среду разработки Visual Studio Code, внешний вид которой представлен на рисунке 19 [24].

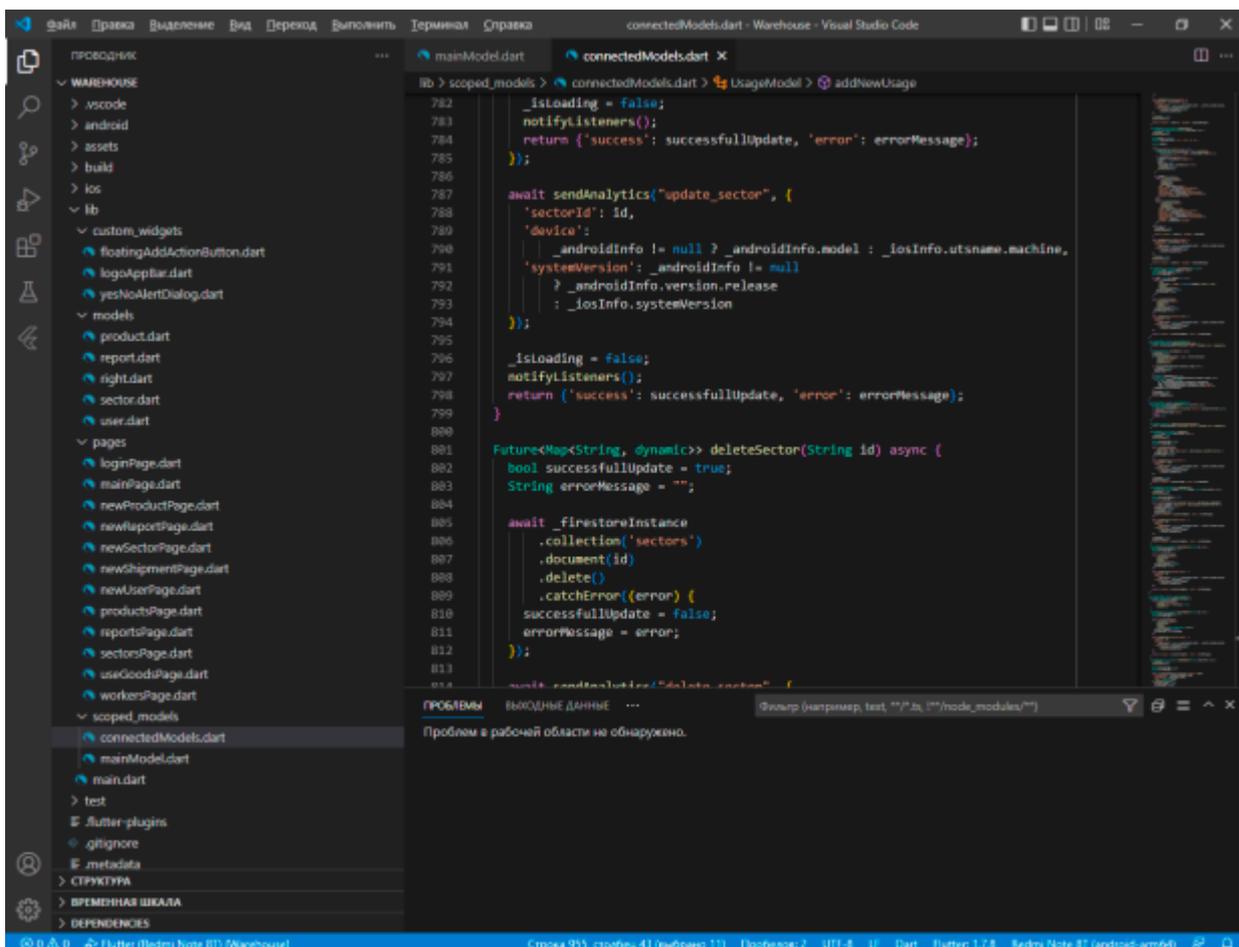


Рисунок 19 – Среда разработки Visual Studio Code

Flutter – относительно недавно появившаяся платформа, но уже активно используется большими компаниями. Сочетает в себе простоту, соизмеримую с программированием web-приложений, но при этом практически не уступает по скорости работы нативным разработкам. Высокая скорость достигается за счет полного отказа от JavaScript и использовании языка Dart, который компилируется в бинарный код и не уступает таким языкам как Swift, Java и Kotlin [17], [19].

Для построения пользовательского интерфейса во Flutter используется декларативный подход, аналогично популярному ReactJS, в основе которого лежат компоненты или виджеты, которые перерисовываются только по необходимости [13].

Перед началом разработки необходимо создать основные директории и файлы проекта, настроить разрешение зависимостей. Они показаны на рисунке 20.

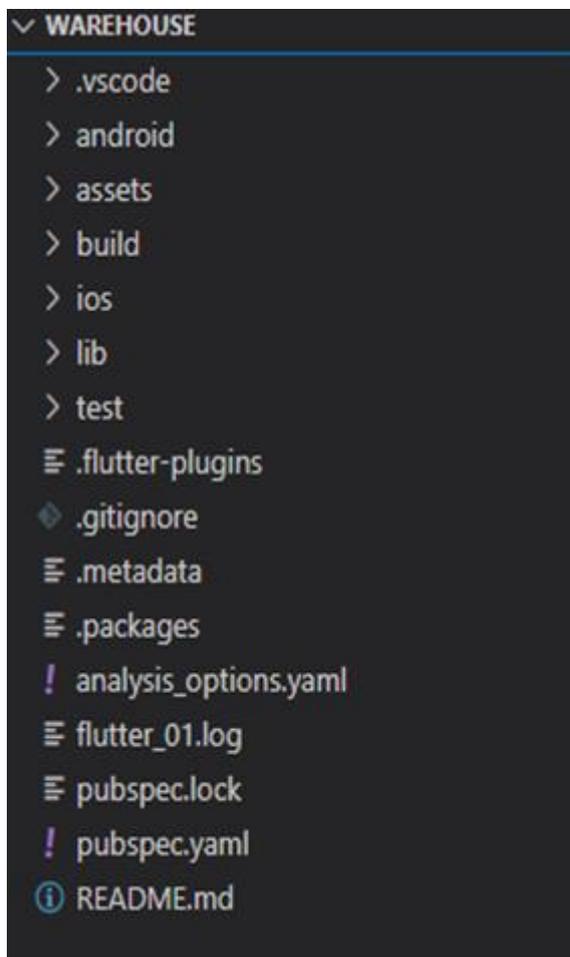


Рисунок 20 – Директории и файлы проекта

Опишем назначение основных директорий более подробно:

- android и ios – содержат ресурсы и описание процесса сборки для соответствующих платформ;
- assets – изображения и другие медиафайлы;
- build – результаты сборки приложения;
- lib – исходный код основных классов на языке Dart;
- test – код тестов.

Также стоит обратить внимание на файл `pubspec.yaml`, содержащий настройки зависимостей. Основные зависимости представлены на рисунке 21.

```
19 dependencies:
20
21   cloud_firestore: ^0.12.9+3
22   firebase_storage: ^3.0.6
23   firebase_auth: ^0.14.0+5
24
25   flutter:
26     sdk: flutter
27
28   cupertino_icons: ^0.1.2
29   scoped_model: ^1.0.1
30   http: ^0.12.0+2
31   grouped_buttons: ^1.0.4
32   image_picker: ^0.4.5
33   barcode_flutter: ^1.0.1
34   firebase_analytics: ^5.0.2
35   device_info: ^0.4.0+2
36
```

Рисунок 21 – Зависимости проекта

Стоит отметить такие зависимости как:

- `cloud_firestore` – подключение облачного сервиса Firestore;
- `firebase_storage` – облачная база данных Firebase Storage;
- `firebase_auth` – авторизация пользователей через Firebase Auth;
- `cupertino_icons` – набор иконок Cupertino;
- `http` – работа с HTTP запросами;
- `image_picker` – выбор изображения;
- `barcode_flutter` – работа со штрихкодами;
- `device_info` – информация об устройстве.

В фреймворке Flutter главным способом создания UI являются виджеты. Они описывают как должно выглядеть представление с учетом конфигурации и состояния. При написании кода необходимо расширить один из базовых классов фреймворка `StatelessWidget` или `StatefulWidget`. Первый не изменяет

свое состояние, второй же имеет такую возможность. В качестве примера на рисунке 22 представлен виджет, отвечающий за авторизацию.

```
8 class LoginPage extends StatefulWidget {
9   final MainModel model;
10
11   LoginPage(this.model);
12
13   @override
14   State<StatefulWidget> createState() {
15     return _LoginPageState();
16   }
17 }
18
19 class _LoginPageState extends State<LoginPage> {
20   final _emailController = TextEditingController();
21   final _passwordController = TextEditingController();
22
23   bool _hidePassword = true;
24
25   @override
26   void initState() {
27     Platform.isAndroid
28       ? widget.model.getAndroidInfo()
29       : widget.model.getIosInfo();
30     super.initState();
31   }
32
33   @override
34   void dispose() {
35     _emailController.dispose();
36     _passwordController.dispose();
37     super.dispose();
38   }
39 }
```

Рисунок 22 – Код виджета авторизации

Получение данных осуществляется путем взаимодействия с облачным хранилищем Firestore. Исходный код приведен на рисунке 23.

```

16 class ConnectedModels extends Model
17     final Firestore _firestoreInstance = Firestore.instance;
18     final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;
19     final FirebaseStorage _firebaseStorage = FirebaseStorage.instance;
20
21     final DeviceInfoPlugin deviceInfo = DeviceInfoPlugin();
22     AndroidDeviceInfo _androidInfo;
23     IosDeviceInfo _iosInfo;
24
25     void getAndroidInfo() async {
26         | _androidInfo = await deviceInfo.androidInfo;
27     }
28
29     User _authenticatedUser;
30     bool _isLoading = false;
31
32     Future<Map<String, dynamic>> uploadImage(
33         String fileName, File _imageFile) async {
34         if (_imageFile != null) {
35             try {
36                 final fileExtension = extension(_imageFile.path);
37                 final StorageReference storageRef =
38                     | _firebaseStorage.ref().child(fileName + fileExtension);
39
40                 final StorageUploadTask uploadTask = storageRef.putFile(_imageFile);
41
42                 final StorageTaskSnapshot taskSnapshot = await uploadTask.onComplete;
43
44                 String _imageDownloadUrl = await taskSnapshot.ref.getDownloadURL();
45

```

Рисунок 23 – Взаимодействие с Firestore

Рассмотрим реализацию бизнес-логики на примере работы с секциями склада. Исходный код класса приведен на рисунке 24, расширяет ранее созданный ConnectedModels.

```

726 class SectorModel extends ConnectedModels {
727     Future<String> getSectorNameById(String sectorId) async {
728         DocumentSnapshot document =
729             await _firestoreInstance.collection('sectors').document(sectorId).get();
730         if (document != null) {
731             return document.data['name'];
732         } else {
733             return null;
734         }
735     }
736
737     Stream<QuerySnapshot> getSectorsFirestoreStream() {
738         return _firestoreInstance.collection('sectors').orderBy('name').snapshots();
739     }
740
741     Future<Map<String, dynamic>> addSector(
742         String name, String description) async {
743         _isLoading = true;
744         notifyListeners();
745
746         bool successfullAdd = true;
747         String errorMessage = "";
748
749         await _firestoreInstance
750             .collection('sectors')
751             .add({'name': name, 'description': description}).catchError((error) {
752                 successfullAdd = false;
753                 errorMessage = error;
754             });
755

```

Рисунок 24 – Реализация бизнес логики работы с секциями

В классе ConnectedModels реализуются следующие методы:

- getSectorsFirestoreStream – взаимодействие с облачным хранилищем;
- getSectorNameById – получение объекта секции по уникальному идентификатору;
- addSector – добавить секцию;
- updateSector – обновить секцию;
- deleteSector – удалить секцию.

3.6 Описание функциональности мобильного приложения

После установки и запуска мобильного приложения открывается форма авторизации, показанная на рисунке 25.

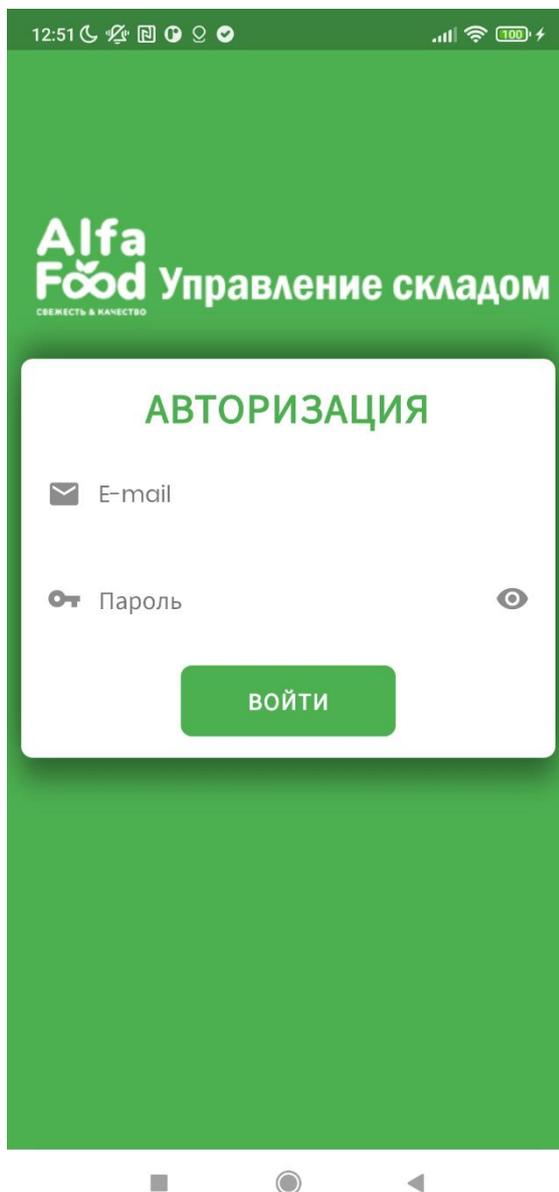


Рисунок 25 – Форма авторизации

После ввода корректных данных для авторизации осуществится переход на главный экран приложения с возможностью отображения меню, интерфейс которого показан на рисунке 26.

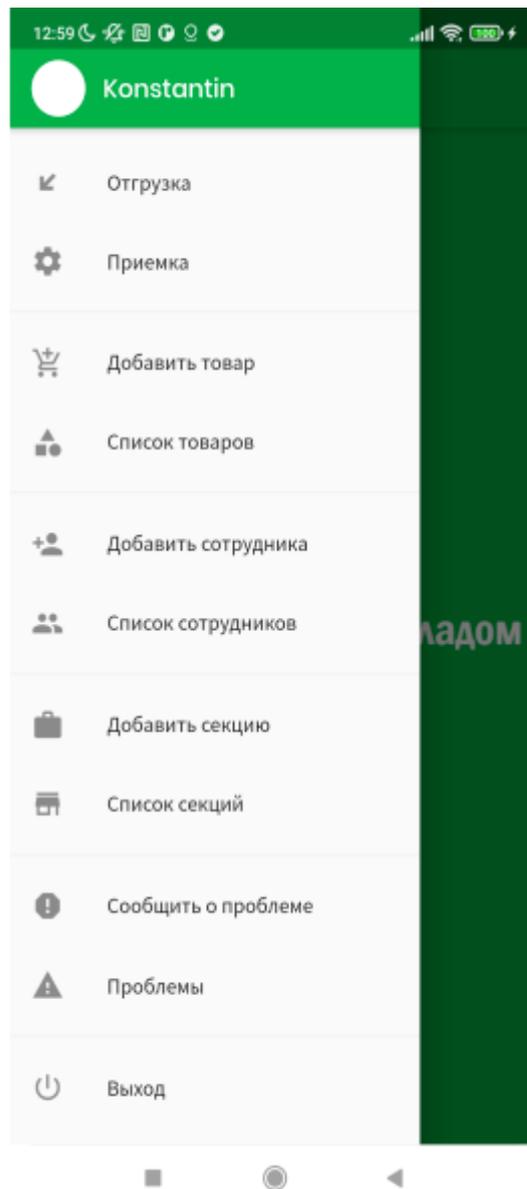


Рисунок 26 – Меню на главном экране

Из главного экрана возможен переход на виджеты отгрузка, приемка, добавить товар, список товаров, добавить сотрудника, список сотрудников, добавить секцию, список секций, сообщить о проблеме, проблемы. А также предоставляется возможность выйти из приложения.

На рисунке 27 показана форма управления отгрузкой товара. Необходимо заполнить номер накладной и добавить товары при помощи кнопки +.

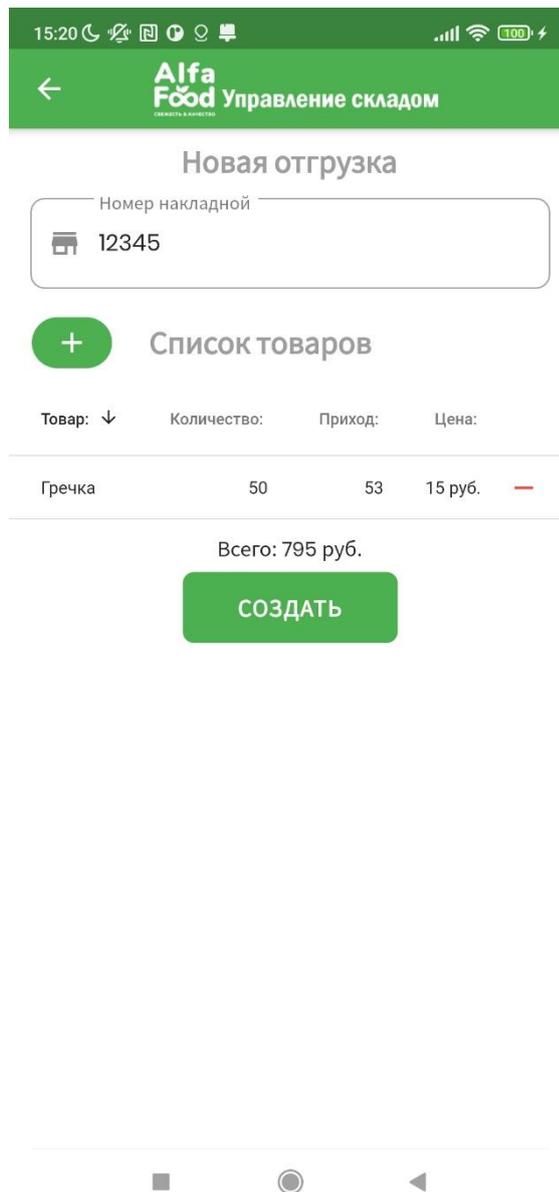


Рисунок 27 – Отгрузка товара

На рисунке 28 показана форма приемки товара. Необходимо ввести номер накладной и добавить товары при помощи кнопки +.

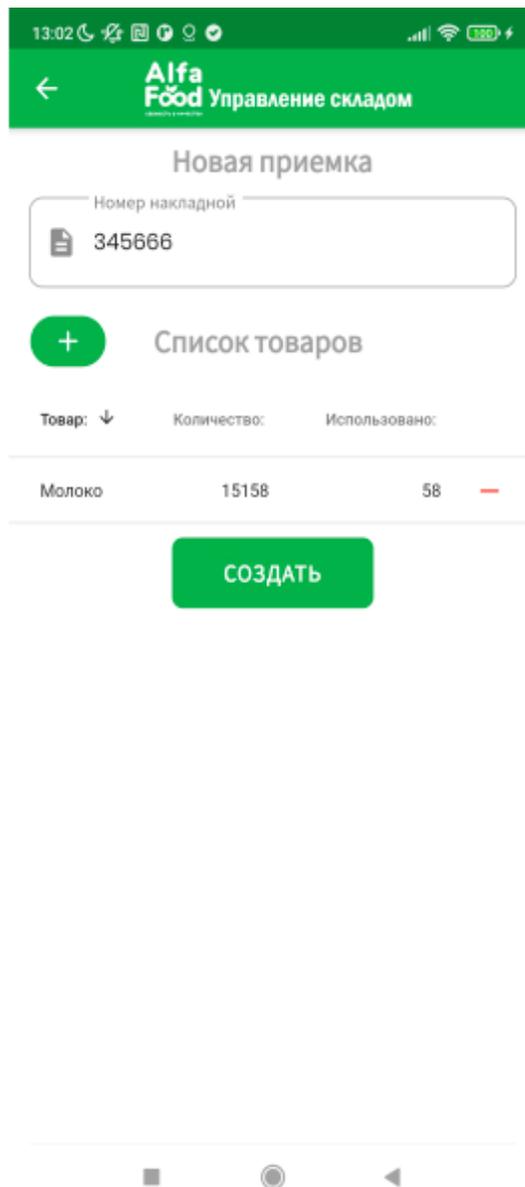


Рисунок 28 – Приемка товара

На рисунке 29 показана форма добавления нового товара. Необходимо заполнить поля название товара, количество, единица измерения, описание, штрих-код, а также выбрать секцию размещения на складе.

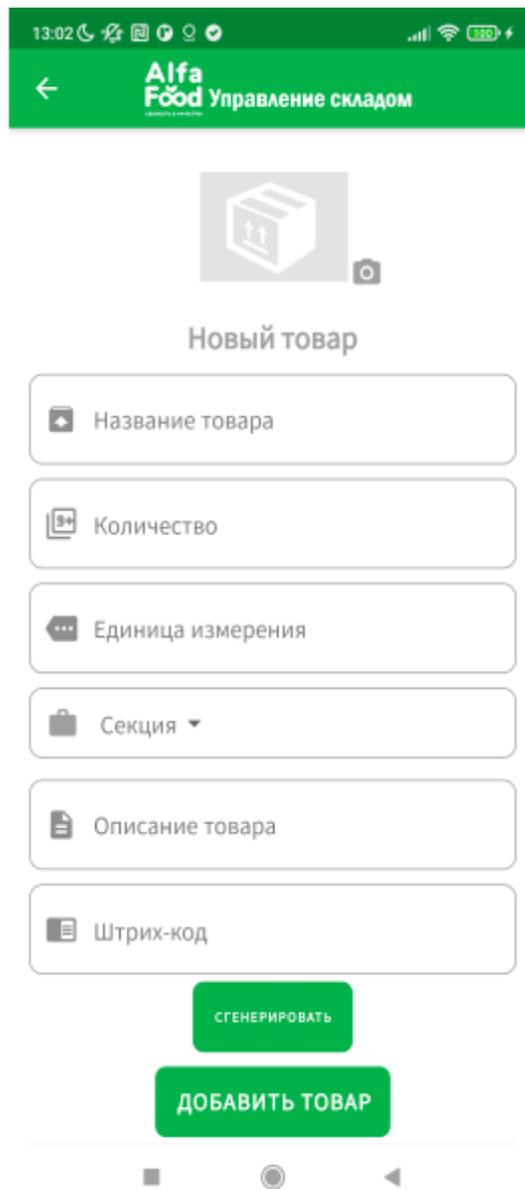


Рисунок 29 – Новый товар

На рисунке 30 показан виджет управления списком товаров. Присутствует возможность редактирования, удаления, а также переход на форму добавления товара.

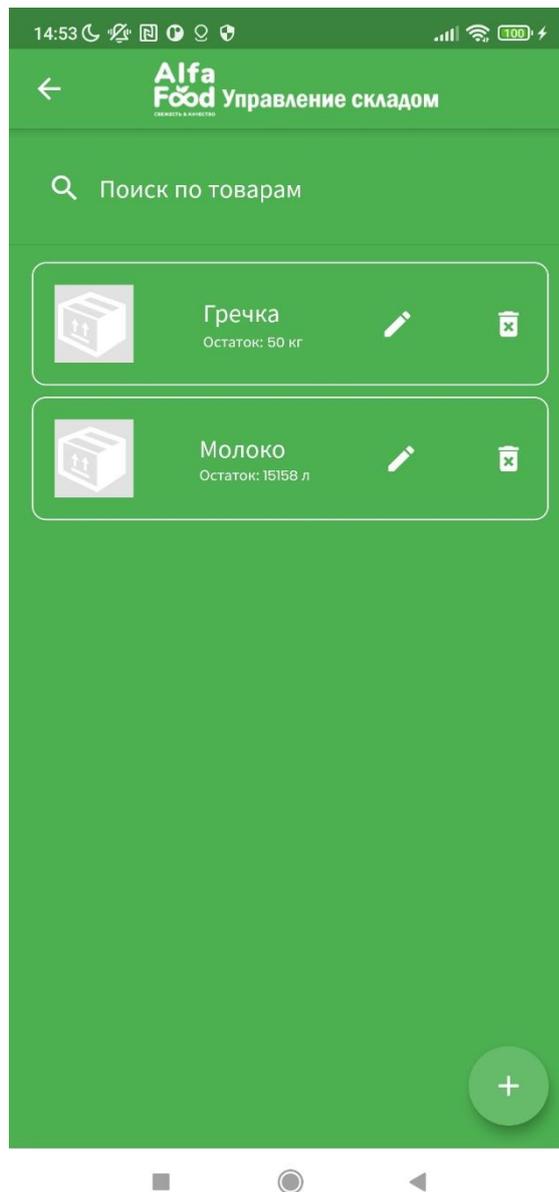


Рисунок 30 – Список товаров

На рисунке 31 показана форма добавления нового сотрудника. Необходимо заполнить поля имя и фамилия, адрес, номер телефона, e-mail, пароль, а также указать права. Admin – заведующий складом, User – кладовщик.

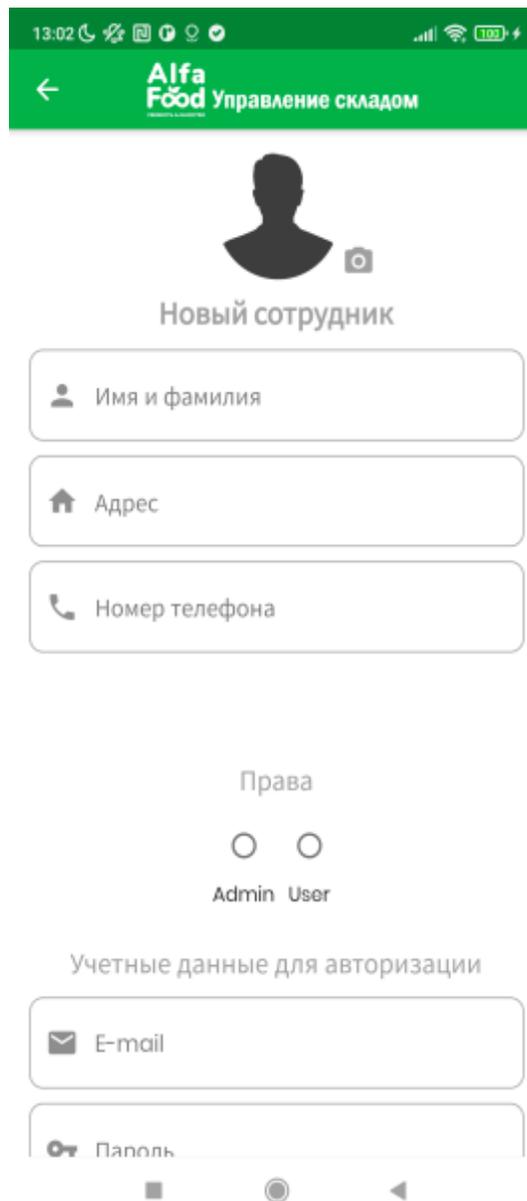


Рисунок 31 – Добавление нового сотрудника

На рисунке 32 представлен виджет списка сотрудников. Присутствует возможность удаления, редактирования, а также переход на форму добавления.

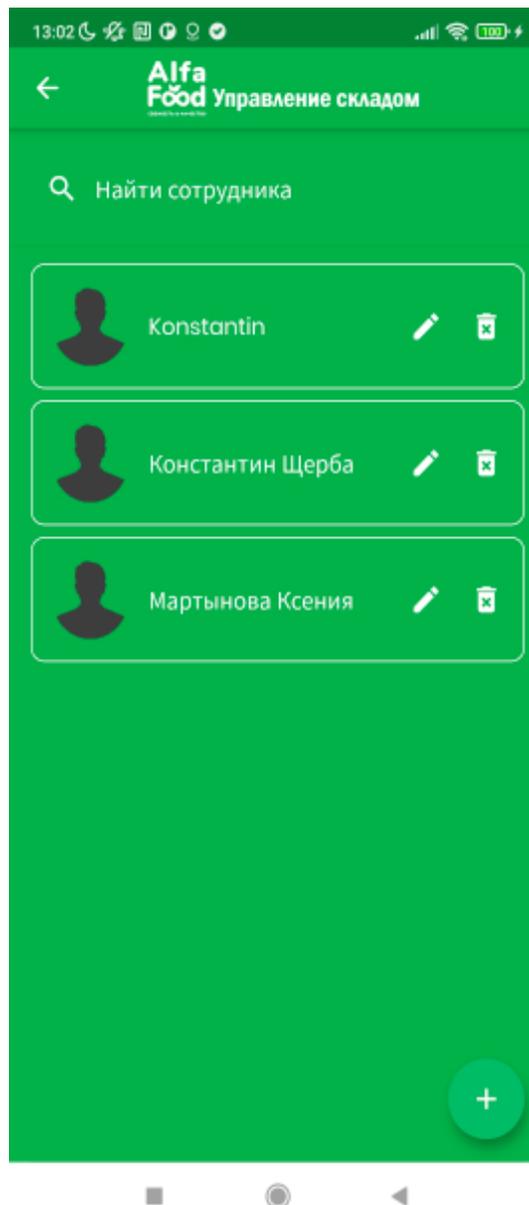


Рисунок 32 – Список сотрудников

На рисунке 33 представлена форма добавления новой секции склада. Необходимо ввести имя секции и описание к ней.

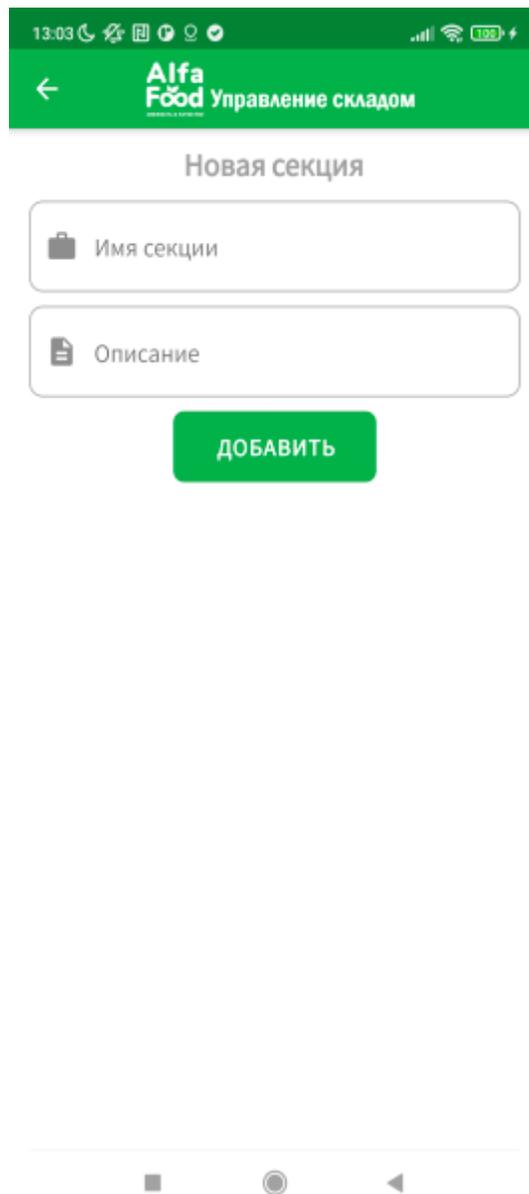


Рисунок 33 – Новая секция

На рисунке 34 представлен виджет списка секций склада. Возможно редактирование, удаление, а также переход на форму создания новой секции.

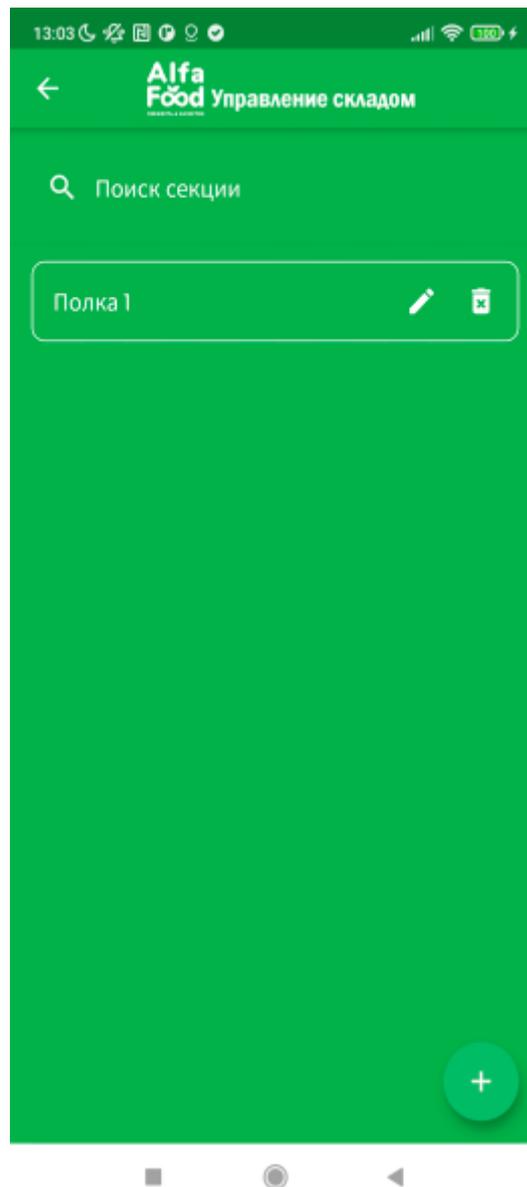


Рисунок 34 – Список секций

На рисунке 35 представлена форма создания новой проблемы. Необходимо описать возникшую проблему и нажать кнопку отправить.

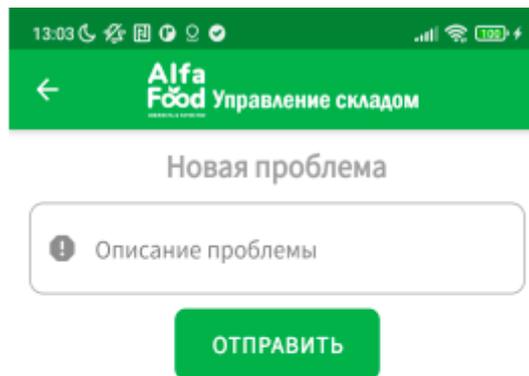


Рисунок 35 – Новая проблема

На рисунке 36 представлен виджет списка проблем. Есть возможность прочитать и удалить запись.

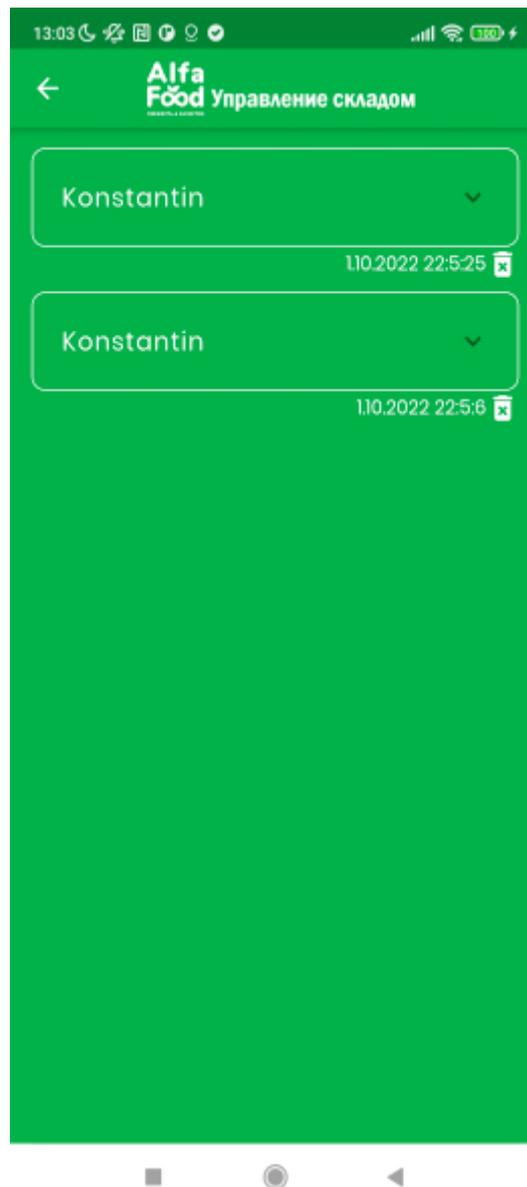


Рисунок 36 – Список проблем

Все вышеперечисленные формы доступны пользователю с правами Admin – заведующему складом. Сотрудник с правами доступа User – кладовщик имеет доступ только к управлению товарами и отправке проблем.

3.7 Оценка и обоснование экономической эффективности разработки мобильного приложения

Рассчитаем технико-экономический эффект от внедрения мобильного приложения для контроля и учета товаров на складе гипермаркета путем

сравнения вариантов с использованием и без использования приложения [2], [3].

На складе компании работает 27 человек, из них доступ к мобильному приложению имеют 1 заведующий складом и 3 кладовщика.

Определим затраты труда в год T_0 (без использования мобильного приложения) и T_1 (с использованием), исходя из среднего количества рабочих дней в месяце – 22 [5]. Данные по суммарным затратам времени за один день возьмем из таблиц 1 и 2:

$$T_0 = 22 \text{ дня} * 12 \text{ мес} * 1200 \text{ мин} = 316800 \text{ мин}$$

$$T_1 = 22 \text{ дня} * 12 \text{ мес} * 600 \text{ мин} = 158400 \text{ мин}$$

Абсолютное снижение затрат в минутах вычислим по формуле (1):

$$\Delta T = T_0 - T_1 = 316800 - 158400 = 158400 \text{ мин} \quad (1)$$

Общий индекс снижения затрат на трудовые ресурсы определим по формуле (2):

$$U_T = \frac{T_0}{T_1} = \frac{316800}{158400} = 2 \quad (2)$$

На рисунке 37 представлена столбчатая диаграмма трудоемкости.

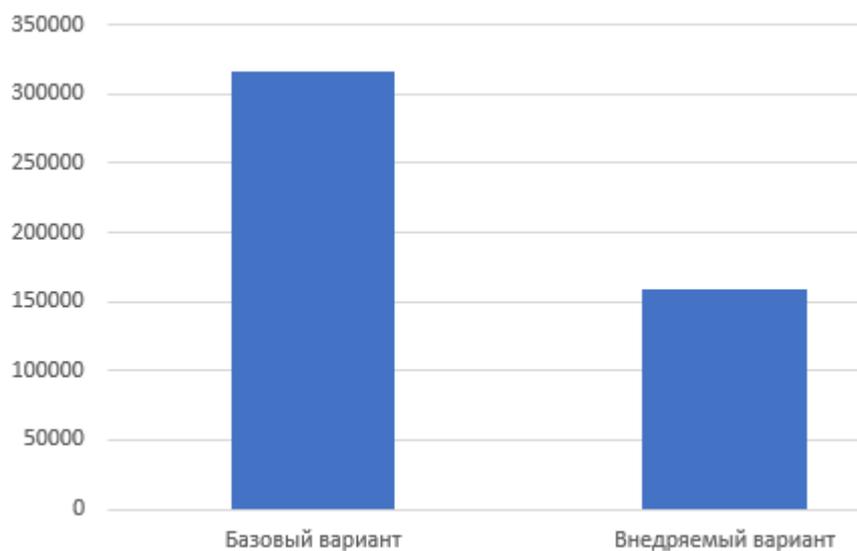


Рисунок 37 – Диаграмма трудоемкости

Зарботная плата заведующего складом составляет 60000 руб., кладовщика – 30000 руб.

Стоимостные показатели:

$$C_0 = 12 * (60000 + 3 * 30000) \text{руб.} = 1800000 \text{руб./год}$$

$$C_1 = 12 * \frac{(60000 + 3 * 30000)}{2} \text{руб.} = 900000 \text{руб./год}$$

Здесь C_0 – стоимость выполнения процессов контроля и учета на складе без использования мобильного приложения, C_1 – с использованием.

Рассчитаем по формуле (3) абсолютное снижение стоимостных затрат:

$$\Delta C = C_0 - C_1 = 1800000 \text{руб.} - 900000 \text{руб.} = 900000 \text{руб./год} \quad (3)$$

На рисунке 38 представлена диаграмма стоимостных затрат.

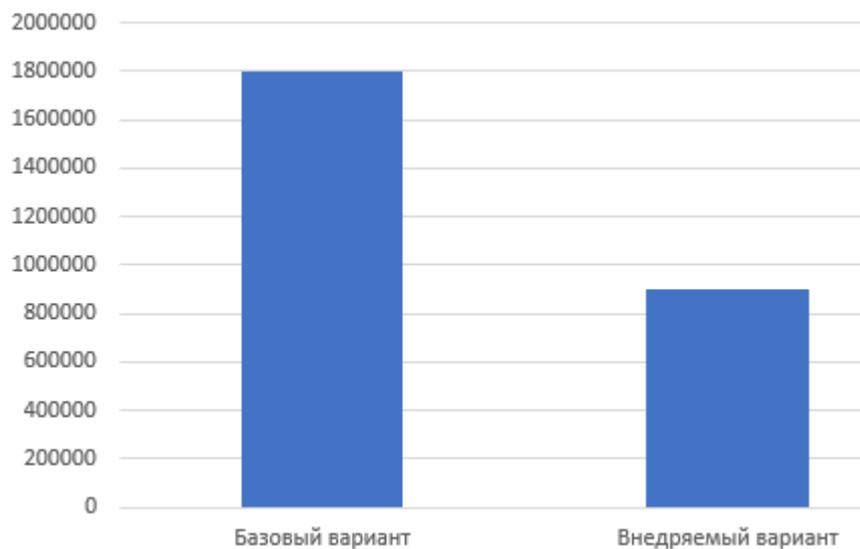


Рисунок 38 – Диаграмма стоимостных затрат

Рассчитаем по формуле (4) индекс уменьшения стоимостных затрат:

$$Y_C = \frac{C_0}{C_1} = \frac{1800000}{900000} = 2 \quad (4)$$

Из расчетов можно сделать вывод, что внедрение мобильного приложения положительно повлияет на функционирование склада гипермаркета ООО «Альфа Фуд».

Выводы по главе 3

При написании главы 3 ВКР была разработана физическая модель мобильного приложения и выбраны технологии разработки ПО и система управления базами данных, описана функциональность и обоснована технико-экономическая эффективность.

Заключение

Целью ВКР является разработка мобильного приложения учета и контроля товара для склада гипермаркета. В приложении реализованы актуальные для ООО «Альфа Фуд» функции авторизации, разграничения доступа, меню, работа со списком товаров, а также его отгрузка и приемка, управление пользователями, секциями склада и проблемами.

Для достижения этой цели была проанализирована деятельность предприятия, спроектирована диаграмма бизнес-процесса склада гипермаркета «как есть», проведен сравнительный анализ аналогов мобильных приложений. В результате выявлена необходимость реализации мобильного приложения, сформированы требования, поставлена задача на разработку и составлена модель «как должно быть».

Получены исходные данные в виде накладных, реестра товарных позиций и отчета об остатках. Разработаны UML диаграммы вариантов использования, классов, вариантов использования и ER-диаграмма при помощи CASE-средства StarUml, благодаря которому удалось сильно облегчить логическое проектирование и на его основе перейти к физическому. Определены требования к аппаратно-программному обеспечению. Полученная в результате логическая модель стала основой для физического проектирования приложения.

Мобильное приложение было разработано с использованием облачной NoSQL-базы данных Firebase, что позволило не покупать выделенный сервер и с легкостью работать с большим объемом документов [20]. Для написания программного кода приложения был применен фреймворк Flutter и язык программирования Dart. Основные преимущества этой связки – поддержка нескольких мобильных платформ, легкость создания экранов и качественный внешний вид приложений даже на старых устройствах. Проведена оценка и обоснование экономической эффективности.

Реализованное мобильное приложение имеет весь необходимый функционал для управления складом гипермаркета. Программный продукт легко справляется со своими задачами. Дополнительное обучение работников склада не требуется, так как интерфейс прост и понятен даже для начинающего пользователя.

Внедрение такого приложения позволит сократить временные затраты и контролировать корректность вводимых сотрудниками склада данных.

В качестве путей развития приложения можно определить введение иерархической структуры секций, что может быть актуально для больших складов, разработку системы фильтров, для более четкого поиска по списку товаров, тем самым расширяя актуальность применения приложения для автоматизации процессов складского учета и контроля.

Цель бакалаврской работы достигнута.

Список используемой литературы и используемых источников

1. Баранова Е. К. Основы информатики и защиты информации : учеб. пособие / Е. К. Баранова. – Москва : ИНФРА-М, 2018. – 183 с. – ISBN 978-5-16-006484-0. – Текст : непосредственный.
2. Баранова И. В. Теория экономического анализа : учеб. пособие / И. В. Баранова, М. А. Власенко, Н. Н. Овчинникова ; под общ. редакцией И. В. Барановой. – Новосибирск : Изд-во НГТУ, 2019. – 170 с. – ISBN 978-5-7782-3912-8. – Текст : непосредственный.
3. Басовский Л. Е. Комплексный экономический анализ хозяйственной деятельности : учеб. пособие / Л. Е. Басовский, Е. Н. Басовская. – Москва : ИНФРА-М, 2021. – 336 с. – ISBN 978-5-16-100198-1. – Текст : непосредственный.
4. Вигерс К. Разработка требований к программному обеспечению. – 3-е изд., допол. / К. Вигерс, Д. Битти. – Санкт-Петербург : БХВ-Петербург, 2014. – 736 с. – ISBN 978-5-9775-3348-5. – Текст : непосредственный.
5. Грэхем Р. Л. Конкретная математика. Математические основы информатики. – 2-е изд. / Р. Л. Грэхем, Д. Э. Кнут, О. Паташник. – Москва : Вильямс, 2021. – 784 с. – ISBN 978-5-8459-1923-6. – Текст : непосредственный.
6. Дейтел П. Android для разработчиков. – 3-е изд. / П. Дейтел, Х. Дейтел, А. Уолд. – Санкт-Петербург : Питер, 2016. – 512 с. – ISBN 978-5-496-02371-9. – Текст : непосредственный.
7. Жаров М. В. Основы информатики : учеб. пособие / М. В. Жаров, А. Р. Палтиевиц, А. В. Соколов. – Москва : Форум, 2017. – 512 с. – ISBN 978-5-91134-232-6. – Текст : непосредственный.
8. Зыкова Г. В. Теоретические основы информатики : учеб. пособие / Г. В. Зыкова, В. В. Пергунов, А. С. Попов. – 2-е изд. – Москва : ФЛИНТА, 2017. – 115 с. – ISBN 978-5-9765-3416-2. – URL: <https://znanium.com/catalog/product/1583195> (дата обращения: 29.07.2022). –

Режим доступа: Электронно-библиотечная система Znanium.com. – Текст : электронный.

9. Иванова В. А. Путь аналитика. Практическое руководство IT-специалиста. 2-е изд. / В. А. Иванова, А. Ю. Перерва. – Санкт-Петербург : Питер, 2018. – 304 с. – ISBN 978-5-496-01679-7. – Текст : непосредственный.

10. Кудинов Ю. И. Основы современной информатики : учеб. пособие / Ю. И. Кудинов, Ф. Ф. Пашенко. – 4-е изд. – Санкт-Петербург : Лань, 2017. – 256 с. – ISBN 978-5-8114-0918-1. – URL: <https://e.lanbook.com/book/91902> (дата обращения: 03.05.2022). – Режим доступа: Электронно-библиотечная система «Лань»; для авторизованных пользователей. – Текст : электронный.

11. Лисицкая Л. В. Совершенствование технологического обеспечения складского учета промышленных запасов и пути совершенствования / Л. В. Лисицкая // Экономика, философия, юриспруденция : Вестник КЭУ. – 2012. – № 1. – С. 194. – URL: <https://www.keu.kz/ru/section-table/31-materialy/1609-2013-01-15-09-18-45.html> (дата обращения 14.05.2022). – Текст : электронный.

12. Майорова Е. С. Современное состояние средств разработки мобильных приложений на платформах iOS, Android, и Windows Phone / Е. С. Майорова, В. А. Ошурков, Л. С. Цуприк // Перспективы науки и образования. – 2015. – № 4. – С. 84. – URL: <https://cyberleninka.ru/article/n/sovremennoe-sostoyanie-sredstv-razrabotki-mobilnyh-prilozheniy-na-platformah-ios-android-i-windows-phone/viewer> (дата обращения 15.05.2022). – Текст : электронный.

13. Медникова О. В. Теоретические основы информатики: учеб. метод. пособие / О. В. Медникова. – Москва : РУТ (МИИТ), 2019. – 56 с. – URL: <https://e.lanbook.com/book/175763> (дата обращения: 03.05.2022). – Режим доступа: Электронно-библиотечная система «Лань»; для авторизованных пользователей. – Текст : электронный.

14. Медникс З. Программирование под Android / З. Медникс, Л. Дорнин, Б. Мик, М. Накамура. – Санкт-Петербург : Питер, 2018. – 496 с. – ISBN 978-459-01115-9. – Текст : непосредственный.

15. Молоков К. А. Основы информатики и программирование под Windows : учеб. пособие / К. А. Молоков. – Москва : Проспект, 2016. – 224 с. – Текст : непосредственный.

16. Потемкина Е. Реализация продукции, работ и услуг : практ. пособие. / Е. Потемкина. – Москва : Лаборатория книги, 2010. – 116 с. – URL: <https://biblioclub.ru/index.php?page=book&id=97014> (дата обращения: 03.05.2022). – Текст : электронный.

17. Роганов Е. А. Основы информатики и программирования : учеб. пособие / Е. А. Роганов. – 3-е изд. – Москва : Изд-во ИНТУИТ, 2021. – 390 с. – ISBN 978-5-4497-0908-0. – Текст : непосредственный.

18. Шлеин В. А. Основы менеджмента : учеб. пособие / В. А. Шлеин, Е. А. Иванова. – Москва : РУТ (МИИТ), 2020. – 135 с. – URL: <https://e.lanbook.com/book/175838> (дата обращения: 03.05.2022). – Режим доступа: Электронно-библиотечная система «Лань»; для авторизованных пользователей. – Текст : электронный.

19. Dart Tutorials : [сайт]. – URL: <https://dart.dev/tutorials> (дата обращения: 02.10.2022). – Текст : электронный.

20. Developer documentation for Firebase : [сайт]. – URL: <https://firebase.google.com/docs> (дата обращения: 02.10.2022). – Текст : электронный.

21. Documentation for app developers : [сайт]. – URL: <https://developer.android.com/docs> (дата обращения: 02.10.2022). – Текст : электронный.

22. Flutter documentation : [сайт]. – URL: <https://docs.flutter.dev> (дата обращения: 02.10.2022). – Текст : электронный.

23. Metanit : [сайт]. – URL: <https://metanit.com/java/android/> (дата обращения: 11.05.2022). – Текст : электронный.

24. Visual Studio : [сайт]. – URL: <https://visualstudio.microsoft.com/ru/>
(дата обращения: 01.10.2022). – Текст : электронный.