

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

Кафедра «Прикладная математика и информатика»
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки, специальности)

Мобильные и сетевые технологии
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка мобильного приложения для работы торгового представителя

Обучающийся

Д.А. Морозов
(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н. Н.В. Хрипунов

(ученая степень, звание, И.О. Фамилия)

Консультант

к.п.н., доцент, С.А. Гудкова

(ученая степень, звание, И.О. Фамилия)

Тольятти 2024

Аннотация

Бакалаврская работа выполнена на тему Разработка мобильного приложения для работы торгового представителя.

Объектом исследования являются процессы работы торгового представителя ООО «Вега-продукт».

Предметом исследования ВКР является разрабатываемое мобильное приложение.

Цель бакалаврской работы состоит в создании функционального мобильного приложения для торгового представителя.

Бакалаврская работа состоит из введения, трёх глав, заключения и списка использованной литературы.

Во введении описывается постановка задач исследования, описывается актуальность работы, кратко излагается общая структура работы.

Первая глава исследования посвящена анализу предметной области и концептуальному моделированию мобильного приложения. В ней формируются требования к мобильному приложению, описываются процессы работы торгового представителя, подлежащие оптимизации.

Во второй главе содержится логическое и функциональное моделирование мобильного приложения. Производится выбор платформы реализации, выстраивается логическая модель приложения и варианты его использования. Также вторая глава содержит модель данных мобильного приложения для последующей реализации.

В третьей главе работы описан процесс разработки мобильного приложения, приведены листинги исходного кода, а также произведено тестирование работы приложения.

В заключении описываются выводы по итогам работы, а также кратко описывается проделанная работа.

Бакалаврская работа выполнена на 65 страницах, содержит 2 таблицы и 28 рисунков.

Abstract

The title of the graduation work is “Development of a mobile application for the work of a sales representative”.

The object of the study is the processes of work of a sales representative of LLC “Vega-product”.

The subject of the research is the mobile application.

The purpose of bachelor's work is to create a functional mobile application for a sales representative.

The bachelor's work consists of an introduction, three chapters, a conclusion and a list of references.

The introduction describes the statement of research objectives, describes the relevance of the work, summarizes the overall structure of the work.

The first chapter of the research is devoted to the analysis of the subject area and conceptual modeling of the mobile application. It forms the requirements to the mobile application, describes the processes of work of a sales representative subject to optimization.

The second chapter contains logical and functional modeling of the mobile application. It selects the implementation platform, builds a logical model of the application and options for its use. Also, the second chapter contains the data model of the mobile application for further realization.

In the third chapter of the work describes the process of development of the mobile application, provides source code listings, as well as testing of the application.

The conclusion describes the conclusions of the work, as well as summarizes the work done.

Bachelor's work is executed on 65 pages, contains 2 tables and 28 figures.

Оглавление

Введение.....	5
Глава 1. Анализ предметной области и функциональное моделирование мобильного приложения	7
1.1 Анализ деятельности торгового представителя на примере ООО «Вега-продукт».....	7
1.2 Возможности оптимизации работы торгового представителя.....	13
1.3 Концептуальное моделирование предметной области исследования.	16
Глава 2. Логическое моделирование мобильного приложения.....	24
2.1 Функциональное моделирование мобильного приложения и выбор платформы реализации.....	24
2.2 Построение логической модели мобильного приложения.....	28
2.3 Разработка модели данных мобильного приложения.....	36
2.4 Выбор средств разработки мобильного приложения.....	37
Глава 3. Реализация мобильного приложения торгового представителя	41
3.1 Разработка программного обеспечения мобильного приложения	41
3.2 Тестирование мобильного приложения.....	56
Заключение.....	60
Список используемой литературы	62

Введение

Взаимодействие торговых представителей с клиентами и управление товарным ассортиментом – это сложный процесс, требующий высокой степени организации и оперативности. От того, насколько эффективно они управляют этими задачами, зависит не только удовлетворенность клиентов, но и успех бизнеса в целом. На примере работы торгового представителя ООО «Вега-продукт» мы рассмотрим актуальность разработки мобильного приложения в контексте оптимизации работы сотрудников за счет предоставления удобного инструмента управления заказами и мониторинга.

Целью данной бакалаврской работы является разработка мобильного приложения торгового представителя.

Актуальность выбранной темы обусловлена сокращением затрат на обслуживание сделок и оптимизацию работы торгового представителя. С помощью внедрения мобильного приложения можно сократить время сотрудника, затрачиваемое на выполнение рутинных задач, и сократит штат специалистов, возложив на каждого имеющегося больше клиентов. За счёт снижения издержек на управление заказами и работу торгового представителя компания получает конкурентное преимущество в виде прямой финансовой экономии, а также лучшего сервиса, чем у конкурентов. Кроме того, мобильное приложение позволит упростить и централизовать передачу данных, минимизируя шансы ошибок за счет человеческого фактора.

Объектом бакалаврской работы являются процессы работы торгового представителя ООО «Вега-продукт».

Предметом бакалаврской работы является разрабатываемое мобильное приложение.

Для выполнения цели бакалаврской работы необходимо выполнить следующие задачи:

- провести анализ деятельности торгового представителя, выделить аспекты, подлежащие автоматизации;

- сформировать требования к мобильному приложению;
- спроектировать архитектуру и логическую структуру мобильного приложения;
- выбрать технологии разработки мобильного приложения и определить целевую платформу реализации;
- реализовать спроектированное мобильное приложение и провести его тестирование.

Выполнение указанных задач позволит спроектировать и реализовать мобильное приложение с учетом всех требований к его функциональности и архитектуре, оптимизировав выбранные процессы работы торгового представителя.

В ходе работы будут применяться методы анализа для выявления основных процессов работы торгового агента и выделения задач, решаемых при помощи мобильного приложения; методы моделирования для составления логических моделей разрабатываемого приложения; методы сравнения для выбора оптимальных инструментов при реализации приложения.

Практическая значимость бакалаврской работы обусловлена возможностью реального использования мобильного приложения в работе торгового представителя и внедрения приложения в существующие бизнес-процессы организации.

Структурно ВКР состоит из введения, трёх глав, заключения и списка использованной литературы.

В первой главе проводится анализ деятельности ООО «Вега-продукт», анализ функций торгового представителя, а также формируются требования к мобильному приложению.

Вторая глава работы посвящена функциональному моделированию мобильного приложения, его логики и архитектуры.

В третьей главе производится реализация мобильного приложения и тестирование его функционала.

Глава 1. Анализ предметной области и функциональное моделирование мобильного приложения

1.1 Анализ деятельности торгового представителя на примере ООО «Вега-продукт»

ООО «Вега-продукт» работает в сфере оптовой продажи продовольственных товаров, предоставляя клиентам широкий ассортимент качественных продуктов. Компания имеет офисы и склады на территории г. Самара и г. Тольятти, однако ведет бизнес по всей Самарской области. Торговые агенты в этой организации играют ключевую роль, осуществляя важные функции по установлению и поддержанию деловых отношений с клиентами. Их профессионализм, навыки в области продаж и стремление к удовлетворению потребностей клиентов являются неотъемлемой частью успеха «Вега-продукт». Торговый представитель несет на себе ответственность за поддержание клиентских отношений, стимулирование продаж и мониторинг конкурентной среды. Его работа напрямую влияет на восприятие бренда и качество обслуживания.

Организационная структура ООО «Вега-продукт» представлена на рисунке 1.

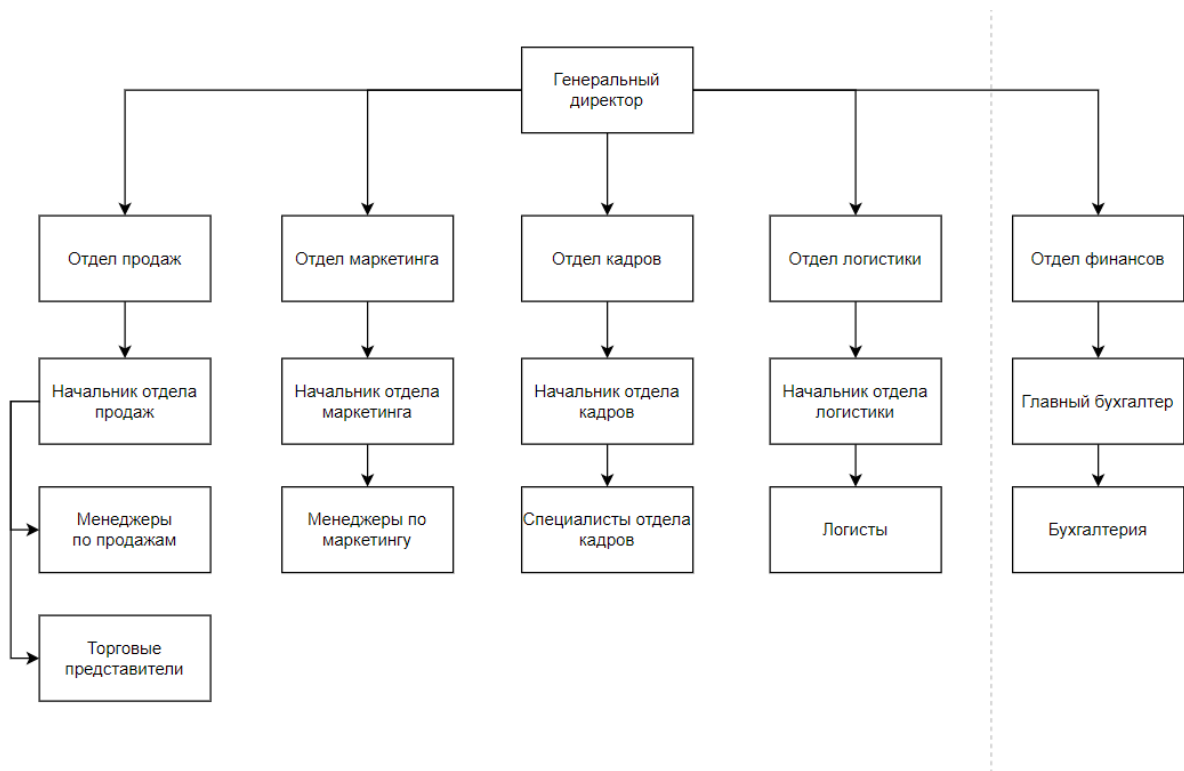


Рисунок 1 – Структура ООО «Вега-продукт»

ООО «Вега-продукт» состоит из пяти основных отделов, и управляется генеральным директором. Каждый отдел возглавляет начальник отдела, в случае отдела финансов – главный бухгалтер. Рассмотрим функции каждого отдела.

Отдел финансов в лице главного бухгалтера отвечает за финансовую стратегию предприятия, контроль денежных потоков, а также бухгалтерию. Сама бухгалтерия занимается финансовым учетом организации, составлением отчетности и ведением всех финансовых операций.

Начальник отдела логистики занимается организацией поставок товаров, их распределением, а также отвечает за товарный склад. Логисты занимаются оперативным управлением складскими запасами, логистикой и доставкой продукции от поставщика до заказчика.

Отдел кадров отвечает за обеспечение предприятия кадровыми ресурсами, разрабатывает кадровую политику организации, обеспечивает

соблюдение прав, льгот и гарантий работников, а также учет работы сотрудников.

Отдел маркетинга в лице руководителя отдела занимается разработкой маркетинговых сценариев и продвижением продукции на рынке. Менеджеры по маркетингу обеспечивают продвижение продукции при помощи рекламы, занимаются анализом рынка и конкурентов.

Отдел продаж – основной источник прибыли предприятия. Именно сотрудники отдела продаж обеспечивают сбыт продукции, занимаются поиском новых клиентов, поддерживают отношения с клиентами и проводят сделки.

С ростом конкуренции и увеличением объемов данных компании сталкиваются с необходимостью совершенствования методов взаимодействия с клиентами, оптимизации процессов принятия решений и повышения эффективности рабочих процессов.

В данном контексте, разработка специализированного мобильного приложения является актуальным шагом для повышения эффективности деятельности торгового представителя. Мобильное приложение может значительно упростить задачи, такие как мониторинг товарного ассортимента, взаимодействие с клиентами, планирование маршрутов и сбор данных [2].

С учетом постоянно меняющихся требований клиентов и конкурентной среды, необходимо подробно разобраться в функциях торгового представителя для дальнейшего совершенствования его работы.

Первой из фундаментальных функций торгового представителя является поддержание отношений с клиентами, встречи с ними, предоставление информации о новинках и услугах, а также оперативную реакцию на запросы и обратную связь. Важно учитывать потребности каждого клиента, их предпочтения, историю покупок и мнение. Это помогает создавать персонализированные предложения, соответствующие индивидуальным потребностям клиентов.

Важным аспектом взаимодействия является также оперативное реагирование на запросы и жалобы клиентов. Торговый представитель должен быть в состоянии предоставлять информацию, решать возникающие вопросы и проблемы, создавая тем самым положительный опыт для клиента. В современной эпохе социальных сетей, где отзывы распространяются мгновенно, важно не только решать проблемы, но и строить положительное восприятие бренда.

Также, торговый представитель должен быть не только продавцом, но и консультантом. Он не только представляет компанию перед клиентами, но и отвечает за установление и поддержание долгосрочных отношений с ними. Ключевыми навыками для торгового представителя являются коммуникабельность, умение убеждать, знание продукции и умение находить подход к разным типам клиентов. Важно быть внимательным к потребностям клиентов и готовым предлагать решения, которые соответствуют их потребностям.

Современные технологии играют существенную роль в оптимизации взаимодействия с клиентами. Эффективное использование CRM-систем (Customer Relationship Management) позволяет отслеживать историю взаимодействия с каждым клиентом, управлять задачами и расписанием визитов, анализировать предпочтения и предсказывать потребности [2]. Это средство позволяет торговому представителю более эффективно управлять клиентской базой и предоставлять более персонализированный уровень обслуживания [3].

Второй основной аспект работы торгового представителя – мониторинг товарного ассортимента. Эффективное управление ассортиментом требует от торгового представителя не только внимания к текущему состоянию товаров, но и способности адаптироваться к изменяющимся предпочтениям и потребностям клиентов.

Одной из ключевых задач торгового представителя является регулярное обновление информации о наличии товаров на складе, включающее

отслеживание остатков, прогнозирование объемов продаж и активное взаимодействие с отделом снабжения для обеспечения постоянной наличности необходимых товаров.

Также торговый представитель должен проводить анализ популярности и рентабельности товаров, с целью выявления тех товаров, которые наиболее эффективно удовлетворяют потребности клиентов и приносят максимальную выгоду компании. Этот анализ позволяет оптимизировать ассортимент, исключая менее востребованные товары и акцентируя внимание на более прибыльных.

Кроме того, торговый представитель отвечает за мониторинг деятельности конкурентов, включая анализ их товарного ассортимента, ценовой политики, а также мероприятий по маркетингу и продвижению. Полученная информация становится важным инструментом для разработки конкурентных стратегий и принятия обоснованных решений по управлению ассортиментом.

Третий пункт в работе торгового представителя – планирование и маршрутизация. Эти элементы играют важную роль в успешной деятельности торгового представителя, особенно в условиях быстро меняющейся деловой среды. Они оказывают прямое влияние на эффективность взаимодействия с клиентами и, в конечном итоге, на результативность всего торгового процесса.

Важным аспектом в планировании является разработка оптимального расписания визитов к клиентам. Торговый представитель должен уметь распределить время для посещения различных точек продаж, учитывая их приоритетность и индивидуальные потребности клиентов. Это требует не только стратегического подхода, но и способности быстро реагировать на изменения в расписании.

Также важна оптимизация маршрутов. Торговый представитель должен учитывать географические особенности районов обслуживания, плотность клиентской базы и возможные транспортные ограничения. С использованием современных технологий и программ для маршрутизации, он может выбирать

оптимальные маршруты, сокращая время в пути и максимизируя время для взаимодействия с клиентами.

Торговый представитель должен уметь оперативно реагировать на неожиданные события, такие как отмены визитов, срочные запросы клиентов или изменения в предпочтениях заказов. Гибкость в планировании позволяет эффективно управлять динамичной рабочей средой [21].

Торговый представитель является первичным источником информации о клиентах, продукции и рынке. В ходе визитов к клиентам он собирает информацию о предпочтениях, объемах продаж, обратной связи и изменениях в потребительском спросе. Этот процесс сбора данных в реальном времени позволяет компании оперативно реагировать на изменения на рынке.

Собранные данные становятся основой для выявления трендов, понимания потребительского поведения и оценки эффективности мероприятий. Анализ данных помогает выделить наиболее успешные продукты, определить слабые стороны стратегии продаж и провести сравнительный анализ с конкурентами [21].

Результаты сбора и анализа данных служат основой для принятия обоснованных решений. Основываясь на полученной информации, торговый представитель может корректировать стратегии продаж, оптимизировать маршруты, реагировать на изменения в потребительском поведении и вносить улучшения в обслуживание клиентов.

Таким образом, сбор и анализ данных представляют собой неотъемлемый элемент работы торгового представителя в современном бизнесе. Они не только обеспечивают текущую информацию для эффективного взаимодействия с клиентами, но и создают фундамент для стратегического управления и принятия обоснованных решений.

Одной из важных задач торгового представителя является разработка и реализация эффективной стратегии продаж. Это включает определение целевой аудитории, анализ конкурентов, позиционирование продуктов на

рынке и разработку индивидуальных подходов к клиентам. На основе этих данных формируется план действий по продажам.

Торговый представитель не просто предлагает товары или услуги, он создает клиентский опыт. Заключение успешных сделок связано не только с предоставлением качественной информации о продуктах, но и с умением эффективно коммуницировать, находить точки соприкосновения с потребностями клиента и предлагать индивидуальные решения.

После заключения сделки торговый представитель продолжает свою работу, следя за выполнением условий контракта, решая возможные проблемы и поддерживая долгосрочные отношения с клиентом. Умение строить и поддерживать отношения является неотъемлемой частью успешной торговой деятельности.

Продажи и заключение сделок представляют собой важную часть работы торгового представителя, требующую высокого уровня профессионализма, стратегического мышления и внимания к потребностям клиентов. Этот процесс требует высоких навыков коммуникации, стратегического мышления и внимательного отношения к потребностям клиентов. Эффективные продажи не только способствуют увеличению доходов компании, но и создают основу для долгосрочных и успешных бизнес-отношений, что является ключевым аспектом в достижении целей предприятия.

1.2 Возможности оптимизации работы торгового представителя

Современные технологии предоставляют торговым представителям средства для оптимизации процессов продаж. Электронные системы оформления заказов, мобильные приложения, CRM-системы и аналитические инструменты облегчают не только работу торгового представителя, но и создают более удобные условия для клиентов. Рассмотрим способы внедрения технологий для оптимизации основных функций торгового представителя.

Первым вариантом оптимизации работы торгового представителя является внедрение CRM-систем. CRM-системы представляют собой эффективные инструменты для оптимизации работы торгового представителя, сосредотачиваясь на управлении взаимоотношениями с клиентами. Реализация CRM-систем в рабочем процессе торгового представителя может значительно улучшить эффективность и результативность его деятельности за счёт следующих функций:

- оптимизация и анализ клиентской базы;
- отслеживание взаимодействия с клиентами;
- инструменты прогнозирования и аналитики;
- управление задачами и планирование.

В первую очередь, CRM-системы необходимы для создания и сохранения информации о клиенте. CRM хранит полную информацию о каждом клиенте в централизованном профиле, например контактные данные, историю покупок, предпочтения, отзывы и другие важные аспекты взаимодействия. Благодаря профилям клиентов, торговые представители могут легко отслеживать историю взаимодействия с каждым клиентом: звонки, электронные письма, встречи и обсуждения. Это помогает понимать предыдущие запросы и предоставлять персонализированный сервис.

Также CRM оснащены аналитическими инструментами, которые помогают торговым представителям анализировать поведение клиентов и прогнозировать дальнейшие тенденции, исходя из истории покупок и поведения клиента. Более точное планирование и прогнозирование позволяет минимизировать риски и повысить общую эффективность продаж.

Кроме того, CRM-системы облегчают коммуникацию внутри команды, позволяя торговым представителям обмениваться информацией, заметками и задачами проектов. Это создает единое информационное пространство для всех участников команды, облегчая планирование и управление задачами.

Альтернативным вариантом оптимизации работы торгового представителя являются мобильные приложения. Ключевое отличие

мобильного приложения от CRM-системы в том, что доступ к мобильному приложению возможен вне зависимости от места работы торгового представителя. Такая особенность обеспечивает гибкость и мобильность в рабочих задачах торгового представителя.

Также мобильные приложения интегрируют функционал электронных торговых платформ, что позволяет торговым представителям проводить транзакции и заключать сделки непосредственно на месте визита к клиенту [11]. Это устраняет необходимость ожидания возвращения в офис для завершения сделок.

Оперативность доступа к данным позволяет использовать мобильные приложения для моментального доступа к актуальной информации в реальном времени. Торговые представители могут быстро получать актуальные данные о ценах, наличии товаров, акциях и других важных аспектах предложения компании.

Ещё одна ключевая функция мобильного приложения состоит в возможности обмена внутренней информацией и автоматизации рабочих задач [13]. Мобильные приложения автоматизируют управление задачами, позволяя торговым представителям создавать, отслеживать и завершать задания прямо из мобильного устройства. Это улучшает управление временем и повышает производительность сотрудников компании.

Возможность интеграции мобильных приложений со сторонними базами данных и облачными платформами гарантирует, что все изменения и обновления информации в централизованном хранилище мгновенно распространятся по всему персоналу, обеспечивая единую картину рабочего процесса.

Мобильные приложения, таким образом, не только обеспечивают доступность информации в любом месте, но и интегрируют торговых представителей в цифровую среду, улучшая оперативность и обеспечивая эффективное ведение бизнеса в условиях постоянных изменений и динамичности рынка.

Для объединения преимуществ CRM-систем и мобильных приложений к реализации предлагается разработать мобильное приложение для торгового представителя с возможностью интеграции с централизованной CRM-системой. Таким образом, удастся совместить мобильность и гибкость в работе торгового представителя с возможностями централизованного управления информацией и использованием инструментов аналитики и прогнозирования. Рассмотрим подробнее, какие моменты в работе торгового представителя возможно автоматизировать при помощи указанных методов.

1.3 Концептуальное моделирование предметной области исследования

Для концептуального моделирования предметной области воспользуемся нотацией IDEF0. IDEF0 (Integration Definition for Function Modeling) – графическая нотация, предназначенная для концептуального моделирования функциональных аспектов системы [14]. IDEF0 предоставляет лаконичную графическую нотацию, что делает представление процессов более понятным для различных участников проекта. Также нотация имеет возможность структурирования процессов с использованием декомпозиции функций на более низкие уровни детализации. Использование IDEF0 обеспечивает базу для анализа и оптимизации функциональных процессов, что помогает в принятии обоснованных решений.

Графическая нотация IDEF0 на практике представляет собой диаграммы функциональных блоков, где каждый блок представляет собой функцию или процесс [8], а стрелки между блоками олицетворяют потоки данных или управления. Основные графические обозначения на диаграммах IDEF0 следующие:

- блоки функций. Блоки, размещенные на диаграмме, представляют функции или процессы системы. Каждый блок имеет уникальное имя и номер, например, A1, A2, B1 и так далее. Многозначные номера

определяют уровень декомпозиции процесса. Например, блок А22 – второй процесс в декомпозиции блока А2;

- стрелки между блоками. Стрелки между блоками указывают на поток данных, материалов или управления между функциональными блоками. От того, с какой стороны блока присоединяется стрелка, зависит роль данных в процессе. Если стрелка присоединяется к блоку слева, это входные данные. Стрелки, выходящие справа из блока – выходные данные или материальные объекты, полученные в результате процесса. Стрелки, входящие в блок сверху – управление процессом (например, руководящие документы и инструкции). Стрелки, входящие в блок снизу – механизмы и ресурсы (в том числе человеческие), используемые в процессе.

Рассмотрим первый уровень приближения бизнес-процессов предметной области. Диаграмма уровня А0 представлена на рисунке 2.

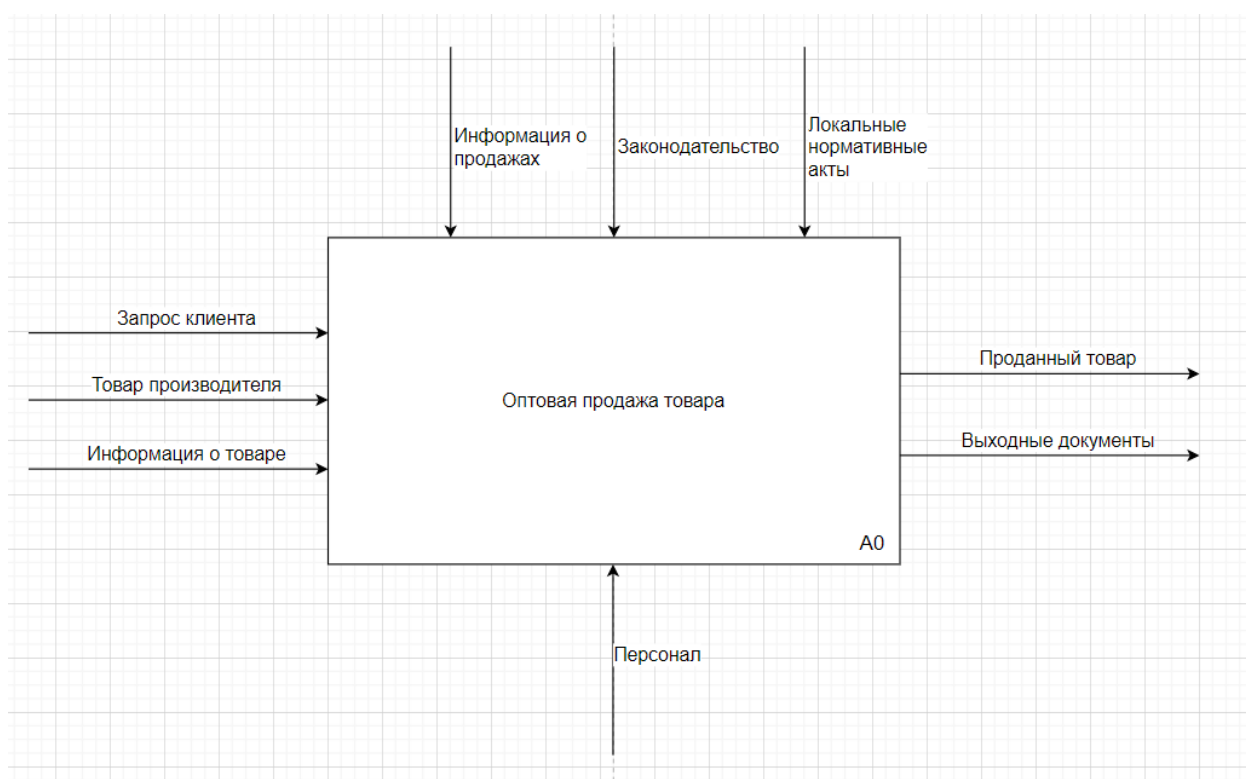


Рисунок 2 – Общая диаграмма процесса оптовой продажи товара «как есть»

Как видно на схеме, исходными данными для продажи товара являются запросы от клиентов, сам товар, а также информация о товаре, полученная от производителя. Процесс оптовой продажи товара регламентирован и зависит от законодательства в сфере торговли, локальных нормативных актов, а также информации о продажах, которая вносит свои коррективы в объёмы и процессы продаж. Основным ресурсом для продаж является персонал организации. Конечным результатом процесса продаж являются проданный товар и выходные документы (счета, акты и т.п). Проведём декомпозицию верхнеуровневого процесса с упором на работу торгового представителя. Декомпозиция процесса приведена на рисунке 3.

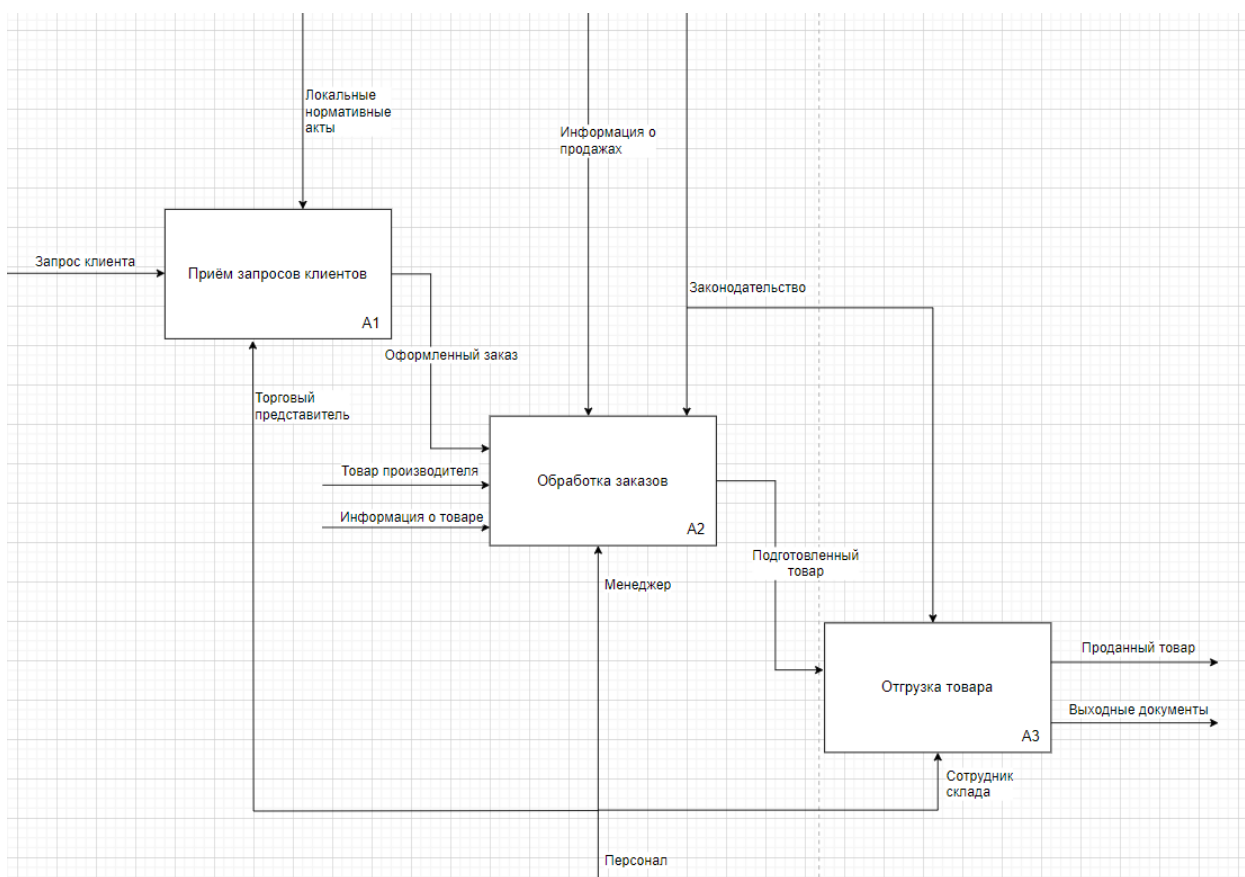


Рисунок 3 – Декомпозиция процесса оптовой продажи товара

Процесс оптовой продажи товара состоит из трёх ключевых этапов: приёма запросов, обработки заказов и отгрузки подготовленного товара

клиенту. Каждым этапом занимается отдельное подразделение персонала, и в контексте данной дипломной работы наиболее интересен этап А1, который и находится в ведении торгового представителя. Проведём его дальнейшую декомпозицию (рисунок 4).

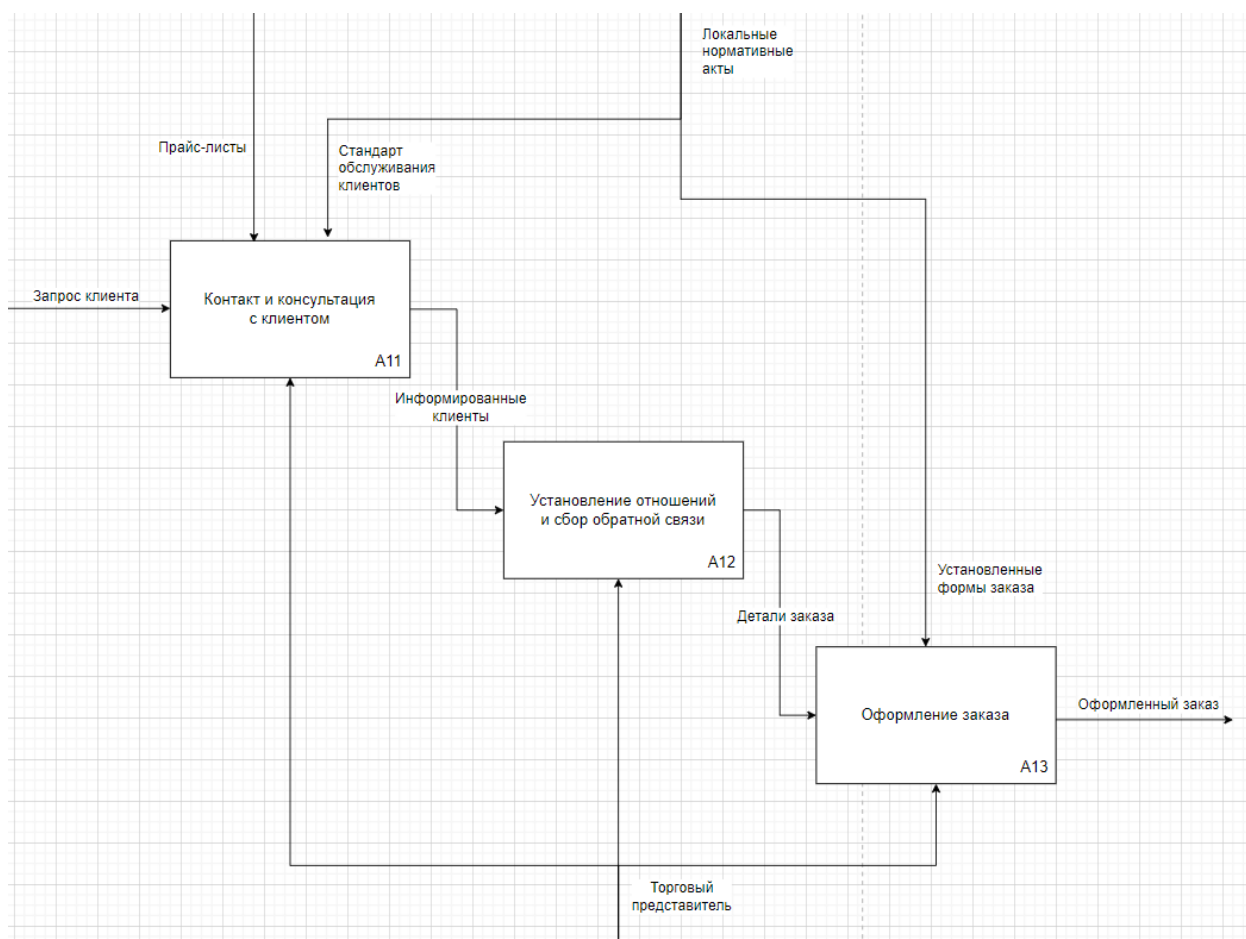


Рисунок 4 – Декомпозиция процесса "Прием запросов клиентов"

В задачи торгового представителя ООО «Вега-продукт» входит установление контакта с клиентом, его консультация по продуктовому ассортименту, установление доверительных отношений и сбор обратной связи по заказу, а также оформление заказа для дальнейшей работы менеджера. В своей работе торговый представитель руководствуется прайс-листами с указанием имеющихся позиций, внутренними стандартами по взаимодействию с клиентами и оформлению заявок.

Разберём подробнее процесс контакта и консультации клиента, как основополагающий для привлечения новых потребителей (рисунок 5).

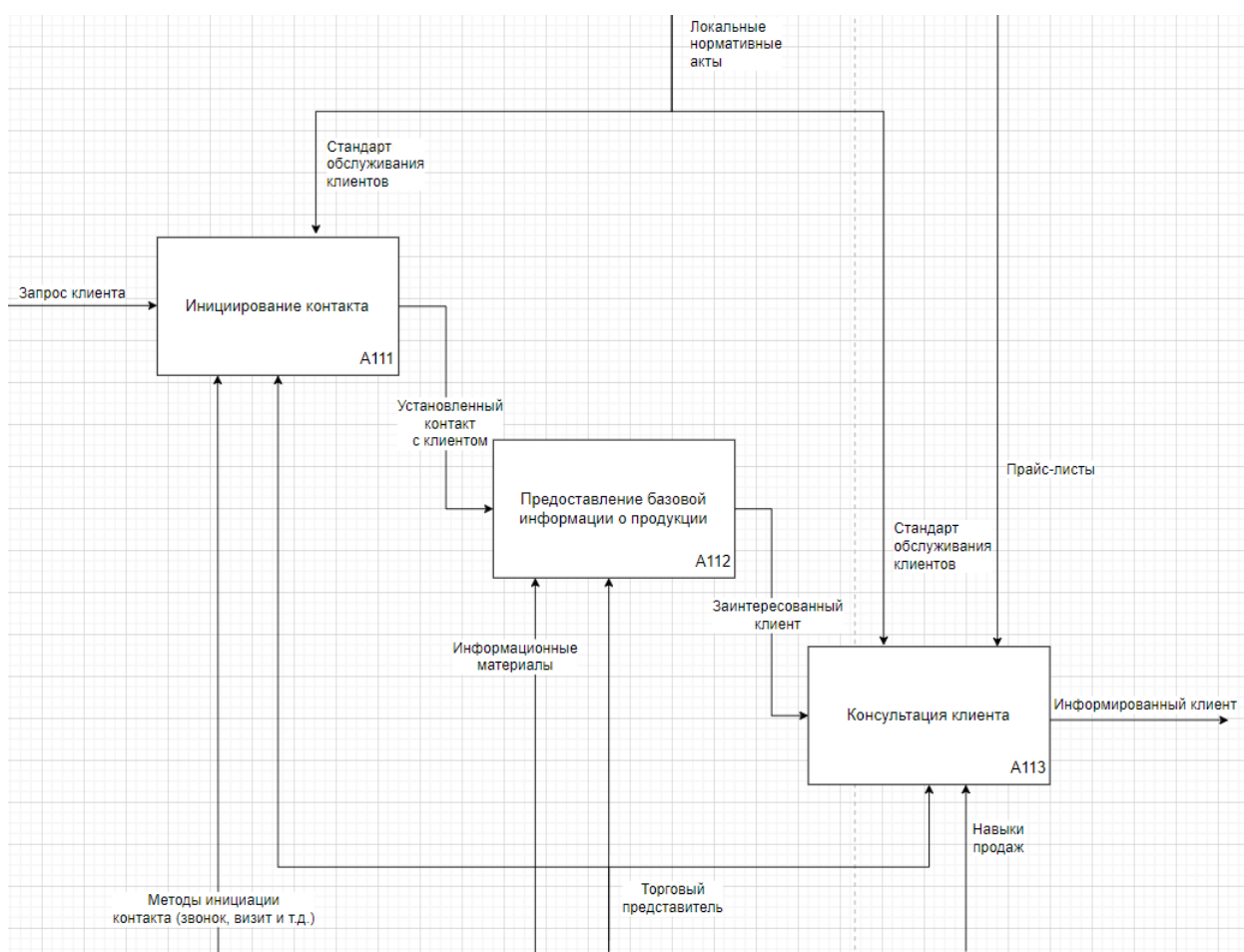


Рисунок 5 – Декомпозиция процесса "Контакт и консультация клиента"

Торговый представитель инициирует контакт с клиентом посредством механизмов, таких как звонок, личная встреча, мессенджеры и т.д. Установив первоначальный контакт с клиентом, торговый предоставляет базовую информацию об ассортименте товаров и их характеристиках, используя при этом имеющиеся информационные материалы. Заинтересовав клиента, торговый представитель проводит консультацию по конкретным позициям, руководствуясь в том числе актуальными прайс-листами, чтобы в итоге сподвигнуть клиента совершить заказ, после чего переходит к процессу A12,

развивая дальнейшие отношения с клиентом и получая обратную связь (рисунок 6).

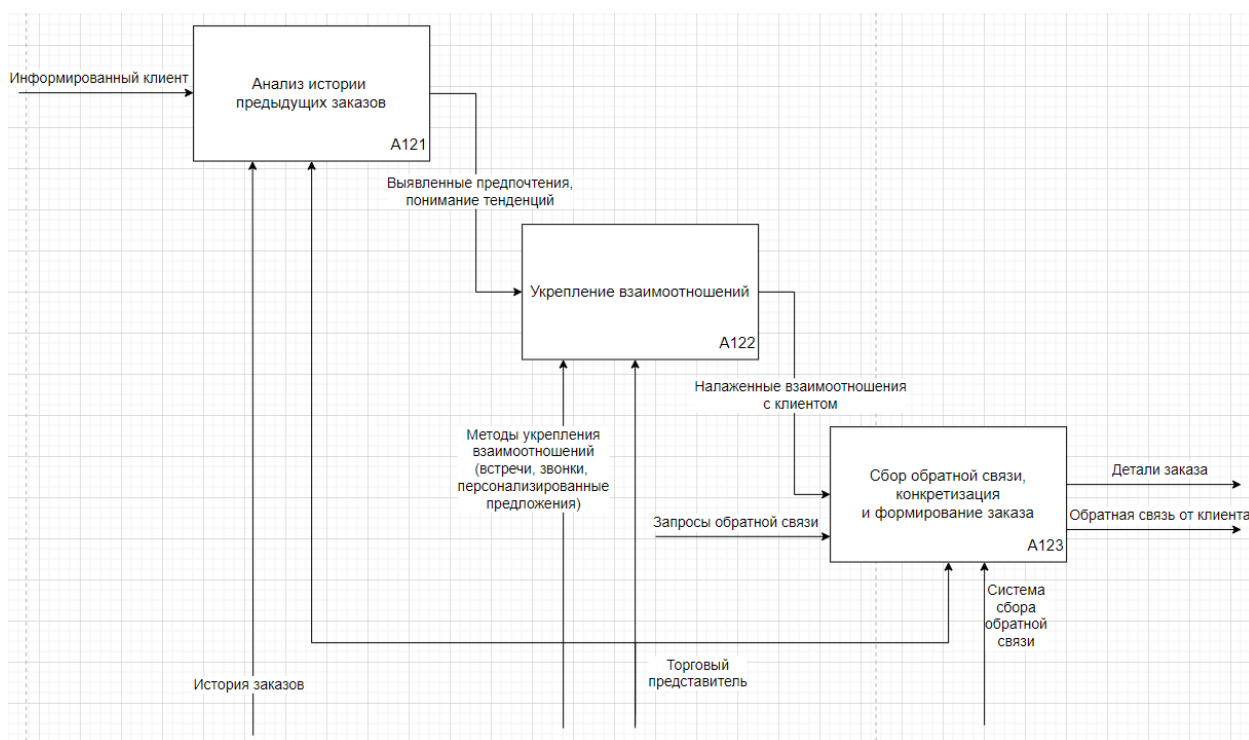


Рисунок 6 – Декомпозиция процесса "Установление отношений и сбор обратной связи"

Установление отношений с клиентом – важная часть работы торгового представителя. В этот этап входит анализ истории предыдущих заказов клиента (если они есть) для выявления предпочтений и тенденций, затем укрепление взаимоотношений путём личных встреч, звонков и подготовки персонализированных предложений на базе выявленных ранее предпочтений (например, в рамках таких предложений торговый представитель может предложить дополнительную скидку или товар на дегустацию). Далее важно собрать обратную связь от клиента для подготовки деталей заказа и улучшения дальнейшего взаимодействия с клиентами. Обсудив детали и получив обратную связь, торговый представитель приступает к этапу оформления заказа (рисунок 7).

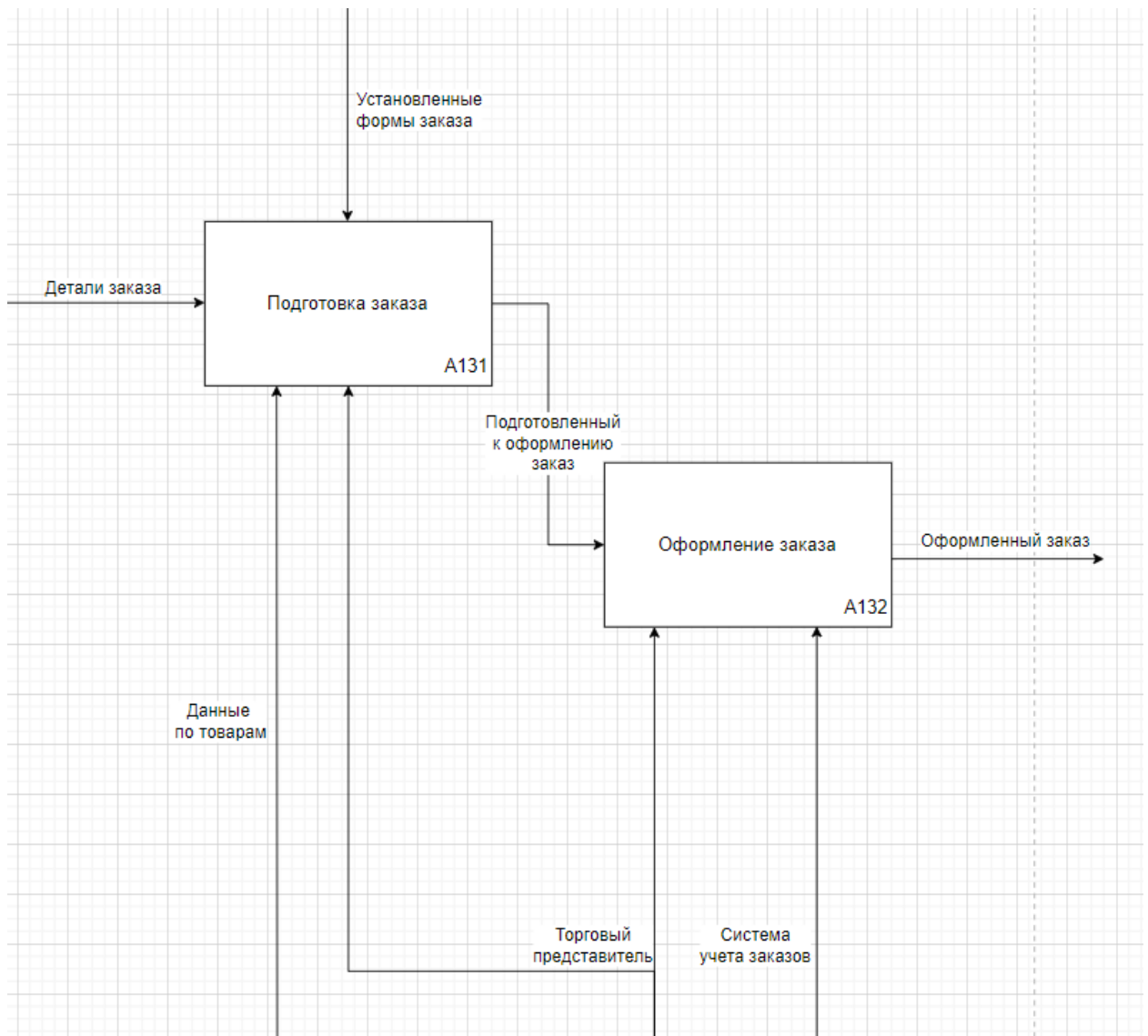


Рисунок 7 – Декомпозиция процесса "Оформление заказа"

Оформление заказа торговым представителем подразумевает приведение заказа к установленной форме и внесение подготовленного заказа в систему учета заказов для дальнейшей передачи менеджеру. При оформлении заказа торговый представитель использует данные по товарам и руководствуется установленными в организации формами оформления.

По результатам концептуального моделирования предметной области мы выявили все этапы работы торгового представителя, и в дальнейшем

можем отталкиваться от приведенных схем для логического моделирования мобильного приложения и выявления схемы оптимизации процессов.

Выводы по главе 1.

В первой главе бакалаврской работы проведен обширный анализ деятельности ООО «Вега-продукт», а также работы торгового представителя. Была приведена структура организации и выявлены аспекты деятельности торгового представителя для дальнейшего исследования.

В ходе концептуального моделирования предметной области и декомпозиции процессов работы торгового представителя были выявлены процессы, которые возможно оптимизировать при помощи мобильного приложения. Декомпозиция процессов производилась при помощи нотации IDEF0, в которой получилось подробным образом описать процессы работы организации. В результате был сформирован перечень процессов для дальнейшего логического моделирования приложения.

Глава 2. Логическое моделирование мобильного приложения

2.1 Функциональное моделирование мобильного приложения и выбор платформы реализации

При формировании функциональных возможностей мобильного приложения мы будем опираться на методологию FURPS+ (Functionality, Usability, Reliability, Performance, Supportability, plus Other Qualities). Методология FURPS+ предоставляет комплексный подход к определению требований к программному обеспечению [5]. Для разработки мобильного приложения торгового агента, рассмотрим основные категории FURPS+ и определим требования в каждой из них (таблица 1).

Таблица 1 - Требования к мобильному приложению

№	Требование	Приоритет	Описание
Functionality (Функциональность)			
1	Хранение профилей клиентов	1	Возможность создания и хранения профилей клиентов с основными данными, историей взаимодействия и предпочтениями.
2	Сегментация и фильтрация	1	Функционал для сегментации клиентской базы и фильтрации данных по различным параметрам, таким как объемы закупок, регион и история покупок.
3	Оптимизация маршрутов	1	Возможность оптимизации маршрутов для визитов к клиентам, учитывая географические параметры, приоритеты и расстояния.
4	Календарь и планировщик задач	1	Функционал для создания и управления календарными событиями, включая визиты, встречи и задачи.
5	Проведение транзакций	2	Возможность заключения сделок и проведения транзакций прямо через мобильное приложение, включая оформление заказов.
6	Интеграция с платежными системами	2	Функционал для интеграции с платежными системами для безопасных электронных транзакций.
7	Интеграция с CRM	2	Автоматическая синхронизация данных с системой управления взаимоотношениями с клиентами (CRM) для обновления клиентской информации в реальном времени.

Продолжение таблицы 1

№	Требование	Приоритет	Описание
8	Доступ к товарному ассортименту	1	База данных о товарах с указанием всех их характеристик и важных параметров, а также возможность отслеживания товарных остатков.
Usability (Удобство использования)			
9	Простота навигации	2	Интуитивная навигация по приложению с легким доступом к основным функциям.
10	Персонализация	5	Возможность настройки интерфейса в соответствии с предпочтениями пользователя.
11	Оптимизация ввода данных	5	Возможность загрузки информации в приложение путем сканирования документов непосредственно в приложении
12	Справка по интерфейсу	4	Возможность вызова контекстной справки, обеспечивающей пользователя информацией по той или иной функции работы приложения
Reliability (Надежность)			
13	Шифрование информации	3	Использование современных методов шифрования для защиты конфиденциальной информации клиентов.
14	Безопасность платежей	1	Гарантии безопасности электронных платежей и защиты финансовых транзакций.
15	Система восстановления при сбоях	2	Возможность быстрого восстановления данных в случае сбоев или потери.
Performance (Производительность)			
16	Оптимизация кода	3	Разработка приложения с учетом оптимизированного кода для максимальной скорости отклика.
17	Кэширование данных	2	Использование механизмов кэширования для ускорения доступа к часто используемой информации.
18	Возможность работы в оффлайн режиме	3	Возможность работать с приложением в оффлайн режиме с последующей автоматической синхронизацией данных.
19	Локальное хранение данных	3	Загрузка и хранение ключевых данных локально для обеспечения работы в условиях отсутствия интернет-соединения.
Supportability (Поддержка)			
20	Техническая поддержка	2	Обеспечение доступа к технической поддержке для решения возникающих вопросов и проблем.
21	Обновления приложения и исправление багов	2	Предоставление регулярных обновлений для добавления новых функций и улучшения стабильности.
22	Совместимость	2	Обеспечение совместимости с различными устройствами и операционными системами.

Продолжение таблицы 1

№	Требование	Приоритет	Описание
Other Qualities (Другие качества)			
23	Система аналитики	3	Возможность сбора данных и создания аналитических отчетов по работе торговых представителей.
24	Возможности масштабируемости	4	Создание мобильного приложения с учетом гибкости для будущего добавления новых функций и масштабирования.
25	Поддержка роста пользовательской базы	3	Гарантии стабильной работы при увеличении числа пользователей.

Для каждого требования указан приоритет реализации от 1 до 5, где 1 – наиболее критичные функции работы мобильного приложения, 5 – наименее критичные функции работы мобильного приложения. Минимальный рабочий продукт должен содержать все требования с приоритетами 1 и 2 для того, чтобы обеспечивать планируемый функционал приложения.

Комбинированный функционал в представленных категориях FURPS+ обеспечит создание мобильного приложения, способного эффективно поддерживать и оптимизировать деятельность торгового агента, предоставляя высокий уровень функциональности, удобства использования, надежности, производительности и поддержки.

Для дальнейшего моделирования мобильного приложения необходимо определиться с платформой его реализации. Выбор платформы для реализации мобильного приложения зависит от ряда факторов, таких как целевая аудитория, бюджет на разработку и прочее. Рассмотрим две основные платформы: iOS и Android.

Операционная система iOS обладает весомым преимуществом в виде собственной экосистемы. Возможности экосистемы Apple позволяют обеспечить кроссплатформенность приложений между различными типами устройств, например iPad, iPhone и Apple Watch [16]. Например, функционал планирования событий и оптимизации маршрутов можно сделать элементом

Apple Watch, а доступ к товарному ассортименту реализовать на планшете iPad для удобства конечного пользователя.

Также iOS обеспечивает безопасность работы с конфиденциальными данными, а интеграция с Apple Pay позволит легко реализовать безопасные транзакции. К сожалению, Apple Pay на данный момент не работает на территории Российской Федерации, однако в будущем ситуация может измениться.

Встроенное шифрование и закрытая операционная система также благоприятно влияют на безопасность приложения, а обязательные к исполнению документы HIG (Human Interface Guidelines) гарантируют удобство использования приложения.

К недостаткам платформы можно отнести ограниченность рынка, поскольку iOS представлена только на устройствах Apple, в то время как Android устанавливается на огромное число устройств. Также приложения iOS разрабатываются на языке Swift, что устанавливает высокий порог вхождения для написания приложений iOS, поскольку Swift – далеко не самый распространенный язык программирования.

Операционная система Android имеет куда больший охват пользовательской аудитории и большее разнообразие устройств. Это даёт возможность реализации универсального приложения для большинства типов устройств, обеспечивая выполнение требований совместимости.

Гибкость в разработке Android-приложений обеспечивается за счёт использования Java и Kotlin в качестве основных языков программирования [17]. Java является одним из самых популярных языков программирования, поэтому порог вхождения для разработчика приложений будет гораздо ниже, чем в случае iOS [18]. Кроме того, Android предлагает открытые инструменты разработки, что также облегчает задачи по реализации приложения [6].

Недостатками Android является большой перечень поддерживаемых версий ОС, что для разработчика означает необходимость многократного тестирования приложения на предмет совместимости с более ранними

версиями. Также различия в аппаратных платформах могут привести к тому, что на устройствах с низкой производительностью приложение не будет работать должным образом.

Основываясь на указанных преимуществах и недостатках, для реализации приложения решено использовать операционную систему Android, как наиболее удобную при разработке и обеспечивающую большую совместимость на аппаратном уровне.

2.2 Построение логической модели мобильного приложения

Логическая модель необходима для визуального представления компонентов разрабатываемой системы. Логическая модель демонстрирует взаимосвязь элементов приложения друг с другом и с пользователем приложения [10]. Благодаря логической модели можно понять, как именно различные компоненты приложения будут функционировать вместе и какими характеристиками обладают те или иные объекты.

В логическом проектировании системы важным этапом является создание диаграммы вариантов использования (use-case diagram). Такая диаграмма описывает, какой именно функционал разрабатываемой системы доступен той или иной группе пользователей (актеров). Диаграмма вариантов использования является частью языка моделирования UML (Unified Modeling Language) – стандартизированного языка моделирования. UML используется для визуализации, проектирования и документирования структуры и поведения программных систем. Он предоставляет набор графических нотаций для описания различных аспектов системы, таких как структура классов, взаимодействие между объектами, последовательность операций и многое другое. Диаграмма вариантов использования разрабатываемого приложения представлена на рисунке 8.

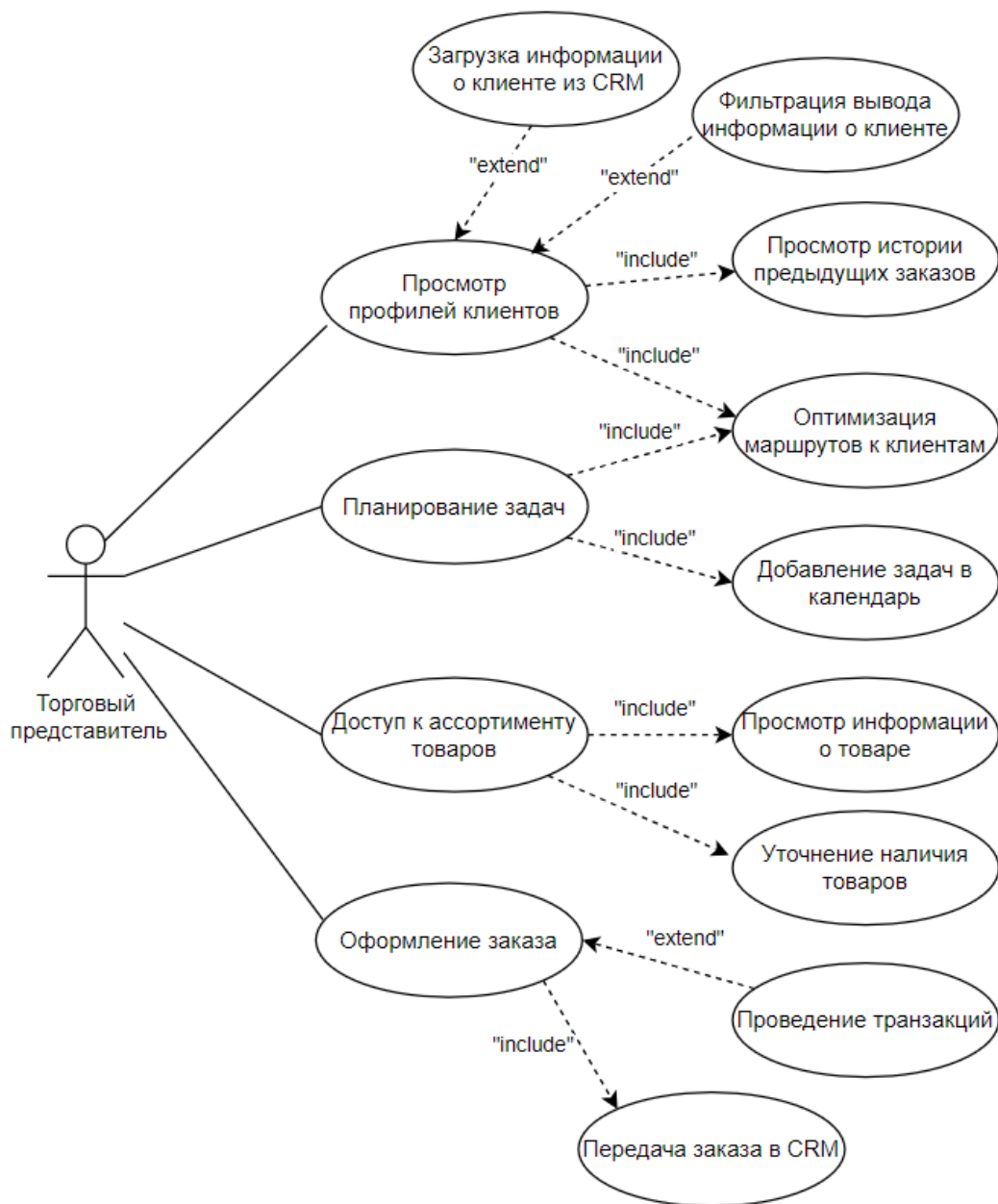


Рисунок 8 – Диаграмма вариантов использования мобильного приложения

Рассмотрим подробнее объекты, представленные на диаграмме. Основным и единственным актором является непосредственно торговый представитель. Приложение не предусматривает использование клиентом или менеджером, поскольку с клиентом работа ведется в присутствии торгового представителя, а менеджер ведёт заказ из CRM-системы.

Основные функции мобильного приложения, доступные торговому представителю – просмотр профилей клиента, планировщик задач, доступ к

ассортименту товаров, а также возможность оформления заказа. Профили клиентов загружаются в мобильное приложение из CRM, мобильное приложение позволяет настроить фильтрацию вывода для визуализации самых необходимых полей для торгового представителя.

Планировщик задач позволяет упорядочить задачи в работе торгового представителя посредством использования календаря, а также позволяет оптимизировать маршруты посредством выгрузки информации об адресе из профилей клиентов.

Доступ к ассортименту товаров необходим для предоставления клиенту информации о товарах, а также уточнения складских запасов по тем или иным позициям товаров.

Также приложение позволяет оформить заказ на месте, внести его в CRM для дальнейшего сопровождения менеджером и сразу оплатить оформленный заказ.

Рассмотрим подробнее прецедент «Планирование задач». Для описания жизненного цикла прецедента воспользуемся диаграммой последовательности. Такие диаграммы используются в UML для демонстрации связей и передачи информации между сущностями в информационной системе в рамках того или иного прецедента [15]. Для диаграмм последовательности характерно использование следующих элементов:

- актер. Как и в диаграммах предыдущего вида, актором именуется субъекты: пользователи, автоматизированные системы или другие подсистемы, которые взаимодействуют с объектами системы;
- объекты. Произвольные сущности информационной системы, обеспечивающие ввод, вывод и обработку информации. Основная задача диаграммы последовательности – показать взаимодействие объектов друг с другом;

- линии жизни. Отображаются на диаграмме пунктирными линиями вниз от объекта и указывают, как долго тот или иной объект существует в контексте процесса;
- фокусы управления. Прямоугольники на линии жизни объекта, которые показывают, в течение какого периода объект выполняет конкретную операцию, отображенную на диаграмме;
- сообщения. Сообщения представляют собой данные, передаваемые между объектами. На диаграмме отображаются при помощи стрелок. Существуют синхронные сообщения, в которых объект-отправитель теряет возможность производить какие-то действия до получения от объекта-получателя ответного сообщения. Такие сообщения отображаются на диаграмме сплошными стрелками с закрашенным треугольником на конце. Ответное сообщение, как правило, содержит данные, необходимые объекту, который ранее передал синхронное сообщение. Ответные сообщения на диаграмме отображаются пунктирными стрелками. Также существует третий тип сообщений – асинхронные сообщения. В случае передачи асинхронного сообщения отправитель не теряет возможности совершать другие действия до получения ответного сообщения. Такие сообщения отображаются сплошной линией с открытой стрелкой.

На рисунке 9 представлена диаграмма последовательности процесса «Планирование задач».

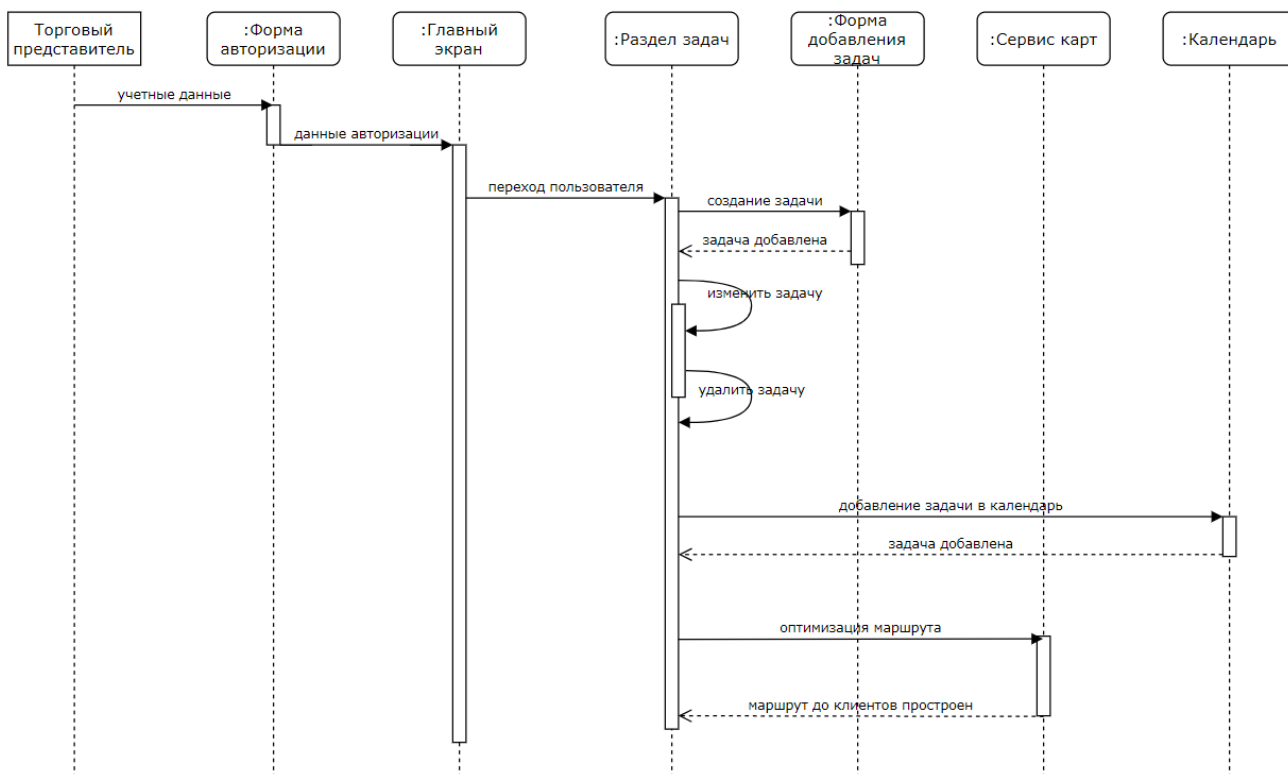


Рисунок 9 – Диаграмма последовательности процесса «Планирование задач»

Для начала взаимодействия с системой пользователь в лице торгового представителя должен пройти процедуру аутентификации, введя свои учетные данные в соответствующую форму. При корректно введенных данных система производит авторизацию пользователя в системе и обеспечивает доступ к главному экрану, из которого можно перейти в раздел задач. В разделе задач пользователю доступно создание новых задач, редактирование или удаление уже имеющихся, а также интеграция задач со сторонним календарем событий и сервисом карт для составления оптимизированного маршрута до клиентов. Добавление задачи осуществляется путем заполнения отдельной формы, в которой пользователь заполняет заголовок, краткую суть задачи, связывает задачу с имеющимся клиентом или заказом, указывает время выполнения задачи и адрес (при необходимости), а также может указать дополнительную информацию о задаче.

На рисунке 10 изображена аналогичная диаграмма для процесса «Просмотр профилей клиентов».

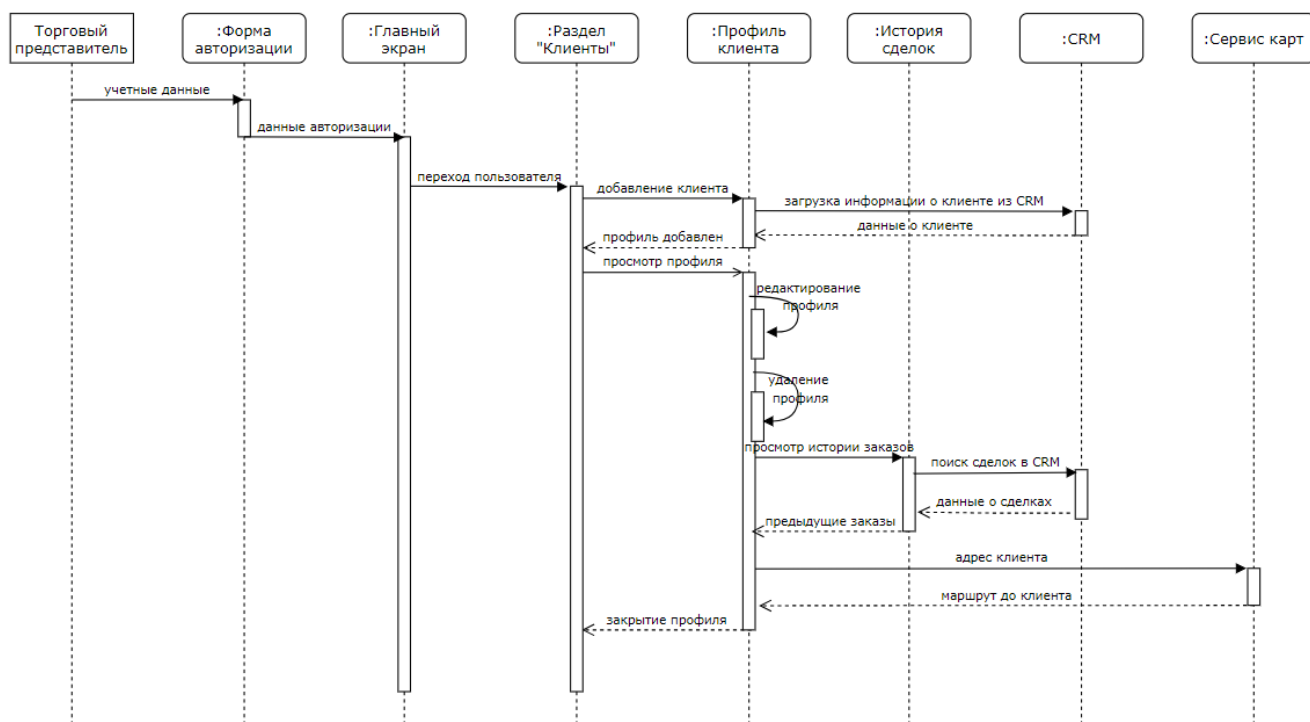


Рисунок 10 – Диаграмма последовательности процесса «Просмотр профилей клиентов»

В этом процессе пользователь, как и в предыдущем, начинает взаимодействие с системой с передачи учетных данных и дальнейшей авторизации. В разделе «Клиенты» пользователь может как добавлять новые профили клиентов, опционально подгружая информацию из CRM-системы (при наличии), так и просматривать существующие профили. При открытии профиля пользователю доступны возможности редактирования профиля, удаления профиля, просмотра истории заказов (также с возможностью выгрузки заказанных позиций из CRM-системы при наличии совершенных сделок), а также возможность составления маршрута до клиента при помощи сторонних картографических сервисов.

Рассмотрим также диаграмму классов проектируемого приложения. Диаграмма классов в UML используется для визуализации структуры информационной системы, описывая классы, их атрибуты, методы и взаимосвязи между ними. Каждый класс на диаграмме классов представляет собой шаблон или чертеж для создания объектов, которые являются экземплярами этого класса. Класс обычно содержит атрибуты, которые определяют его состояние, и методы, которые определяют его поведение. Атрибуты могут быть переменными, свойствами или полями класса, а методы могут быть функциями или процедурами, определяющими операции, которые класс может выполнять.

Взаимосвязи между классами на диаграмме обычно представлены в виде ассоциаций, которые показывают, как один класс взаимодействует с другими. Ассоциации могут быть однонаправленными или двунаправленными, и могут иметь различные мультипликаторы, указывающие количество объектов одного класса, связанных с объектом другого класса. Диаграмма классов также может включать различные виды отношений, такие как обобщение (наследование), реализацию интерфейсов, агрегацию и композицию, которые позволяют описывать более сложные отношения между классами.

Диаграмма классов проектируемого приложения представлена на рисунке 11.

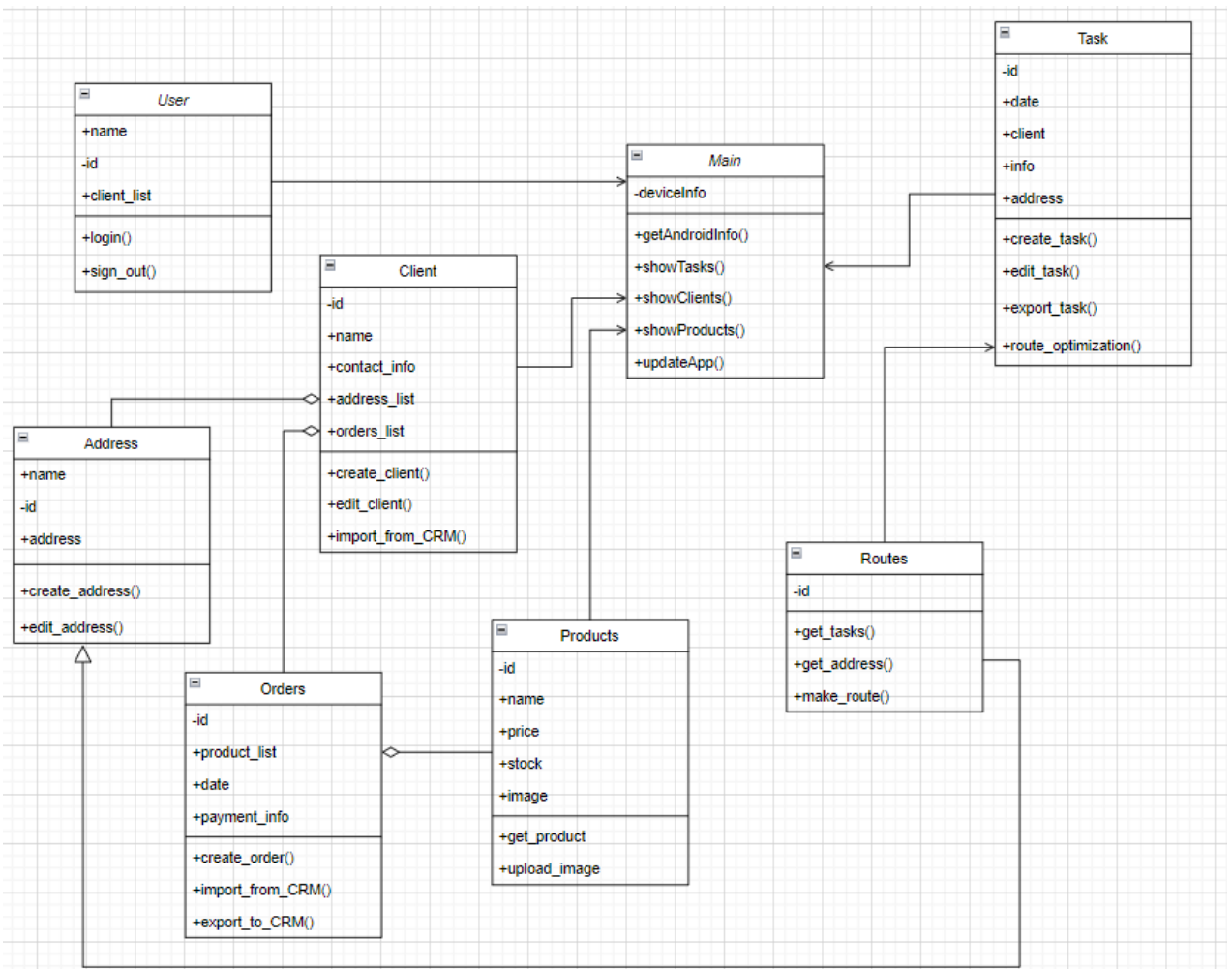


Рисунок 11 – Диаграмма классов приложения торгового агента

Диаграмма классов содержит ключевые объекты проектируемого приложения. Сущность Main связывает основные функциональные возможности приложения, обеспечивая пользователю доступ к различным его частям. Сам пользователь приложения отражен в виде сущности User. Клиенты, прикрепленные к торговому представителю, выражены сущностью Clients. В информации о клиентах также доступны адреса, используемые для оптимизации маршрутов, а также заказы с возможностью создания нового заказа из приложения и загрузки предыдущих из CRM. Задачи торгового представителя выражены сущностью Tasks, для которой доступен экспорт во внешний календарь.

Вышеприведенные диаграммы составляют основу логической модели разрабатываемого приложения [12].

2.3 Разработка модели данных мобильного приложения

Модель данных мобильного приложения представляет собой структурированное представление данных, которые приложение использует для работы. Проектирование модели данных приложения позволяет обеспечить согласованность и правильную организацию данных до этапа непосредственно разработки приложения, предотвращая дальнейшие ошибки и проблемы с данными [24]. Кроме того, правильно спроектированная модель данных мобильного приложения предусматривает возможности масштабируемости и дальнейшей аналитики [10]. В качестве концептуальной модели данных мы можем использовать ранее разработанную диаграмму классов, поскольку сущности данных соответствуют классам мобильного приложения. Далее необходимо определить атрибуты данных и их отношения друг с другом.

Для отображения модели данных используют ER-диаграммы (Entity Relationship model, модель «сущность – связь»). ER-диаграмма состоит из сущностей (объектов), которые составляют основу базы данных. Каждая сущность имеет набор атрибутов, которые описывают их свойства или аспекты. Сущности воздействуют друг с другом посредством тех или иных связей. Основные виды связей в ER-диаграммах: один к одному, один ко многим, многие ко многим [7]. Связи типа «один к одному» подразумевают, что запись одной сущности связана с записью другой сущности. Например, одному продукту соответствует один уникальный номер, и этот номер также привязан к этому продукту [9]. Связь типа «один ко многим» используется в случаях, когда один атрибут сущности связан с несколькими другими. Например, компании принадлежит несколько номеров телефонов, но каждый номер телефона принадлежит только одной компании. Связь вида «многие ко

многим» используется в случаях, когда нескольким атрибутам сущности соответствует несколько атрибутов другой сущности [20]. Предположим, компания имеет несколько филиалов на различных адресах, но кроме этой компании по указанным адресам располагаются и другие организации (например, в торговом центре).

ER-диаграмма формирования заказов в мобильном приложении представлена на рисунке 12.

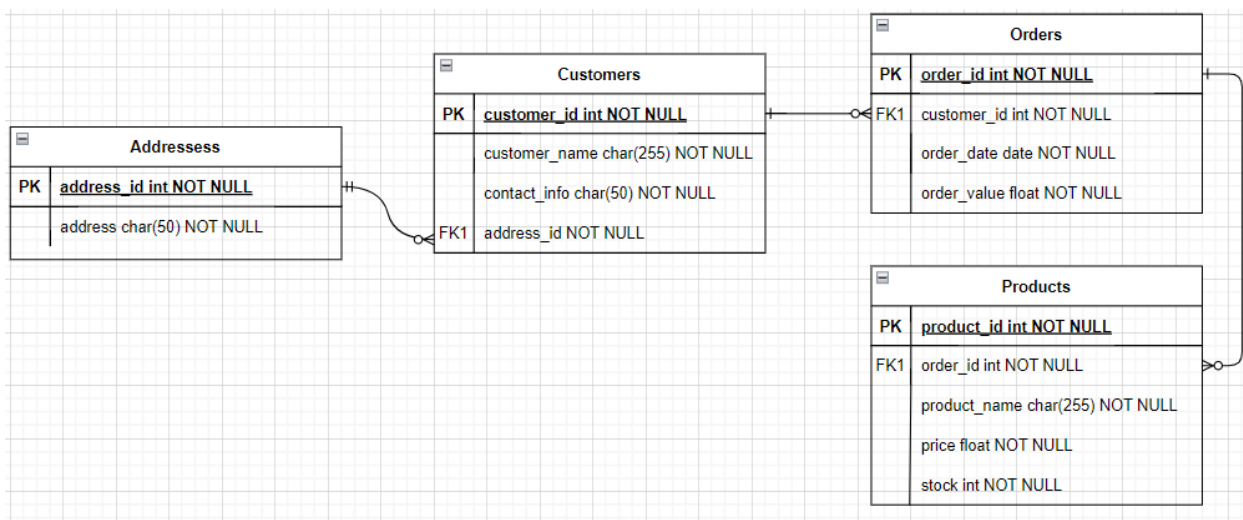


Рисунок 12 – Диаграмма данных создания заказа

Таким образом, отталкиваясь от созданных логических моделей приложения, диаграмм классов и данных, мы можем приступить к реализации проектируемого мобильного приложения. Для этого необходимо определить, при помощи каких средств будет производиться разработка.

2.4 Выбор средств разработки мобильного приложения

Ранее были рассмотрены преимущества и недостатки платформ реализации мобильного приложения и в результате анализа была выбрана платформа Android как наиболее оптимальная для проектируемого приложения. Для операционной системы Android существуют различные

инструменты разработки приложений, которые отличаются как функциональными возможностями, так и особенностями подхода к проекту.

Наиболее популярной средой разработки для Android является Android Studio. Она основана на популярной IDE IntelliJ IDEA и предоставляет разработчикам инструменты для написания кода, отладки, тестирования и профилирования приложений для устройств на базе Android [4].

Основными преимуществами данной IDE являются высокая производительность за счет оптимизации встроенных инструментов разработки, нативная интеграция с Android SDK, обеспечивающая простой доступ к функциям Android [4], которые используются при разработке приложений, а также поддержка множества языков программирования: Android Studio позволяет писать код на Java, Kotlin и C++, что снижает порог вхождения и позволяет реализовать функционал с применением большей универсальности, чем у других IDE.

Кроме того, Android Studio имеет обширный набор встроенных шаблонов реализации и большое сообщество пользователей, готовых при необходимости помочь советом начинающему разработчику.

Недостатками Android Studio являются требовательность к аппаратным ресурсам и сложность первоначальной настройки среды разработки. Несмотря на это, Android Studio официально рекомендуется Google как основная среда разработки под Android.

Также популярной средой разработки Android-приложений является непосредственно IntelliJ IDEA. Данная IDE также поддерживает несколько языков программирования, имеет преимущества в виде быстрой отладки кода и автозаполнения методов, удобный и дружелюбный интерфейс, а также встроенный рефакторинг кода [25].

Функционал IDE дорабатывается посредством плагинов, как официальных, так и созданных энтузиастами сообщества [1]. Недостатками IDE являются большой объем и требовательность к ресурсам компьютера,

ограниченность функционала в бесплатной версии IDE, а также проблемы с совместимостью библиотек на недавно вышедших версиях Android.

Классической бесплатной средой разработки Android-приложений также является продукт некоммерческой компании Eclipse Foundation под названием Eclipse [4].

Среди достоинств этой IDE можно выделить полный перевод на русский язык, что является плюсом для разработчиков из России, хорошую оптимизацию и нетребовательность к ресурсам компьютера в противовес продуктам IntelliJ, а также огромные возможности для расширения функционала посредством использования плагинов.

К сожалению, с 2014 года Eclipse потеряла статус официально поддерживаемой среды разработки, вследствие чего имеет проблемы с совместимостью на новых версиях Android.

Таким образом, наилучшей средой для разработки приложения торгового представителя является IDE Android Studio, как наиболее оптимизированное решение с возможностями для последовательного тестирования кода и шаблонами самых популярных функциональных требований.

Выводы по главе 2.

Во второй главе бакалаврской работы было проведено логическое моделирование мобильного приложения.

На этапе функционального моделирования были выявлены требования к приложению по модели FURPS+, описывающие требования к функциональности мобильного приложения и другим его аспектам. Исходя из требований, платформой реализации мобильного приложения была выбрана ОС Android, как наиболее оптимальная по доступности и удобству использования.

Также была построена логическая модель мобильного приложения и разработаны диаграммы вариантов использования и диаграммы последовательности процессов в приложении. Для диаграмм

последовательности были выбраны наиболее частые с точки зрения работы торгового представителя функции – работа с задачами, а также работа с клиентами.

Эти диаграммы демонстрируют, как именно пользователь может взаимодействовать с приложением и какие функции доступны в тех или иных сценариях.

Диаграмма классов мобильного приложения необходима для того, чтобы логически выстроить связи между классами при последующей их реализации в среде разработки.

Кроме того, была выбрана среда разработки мобильного приложения. Выбор производился между Android Studio, IntelliJ IDEA и Eclipse.

По итогам сравнительного анализа было решено реализовать мобильное приложение в IDE Android Studio.

Глава 3. Реализация мобильного приложения торгового представителя

3.1 Разработка программного обеспечения мобильного приложения

Интерфейс Android Studio состоит из нескольких основных элементов и представлен на рисунке 13.

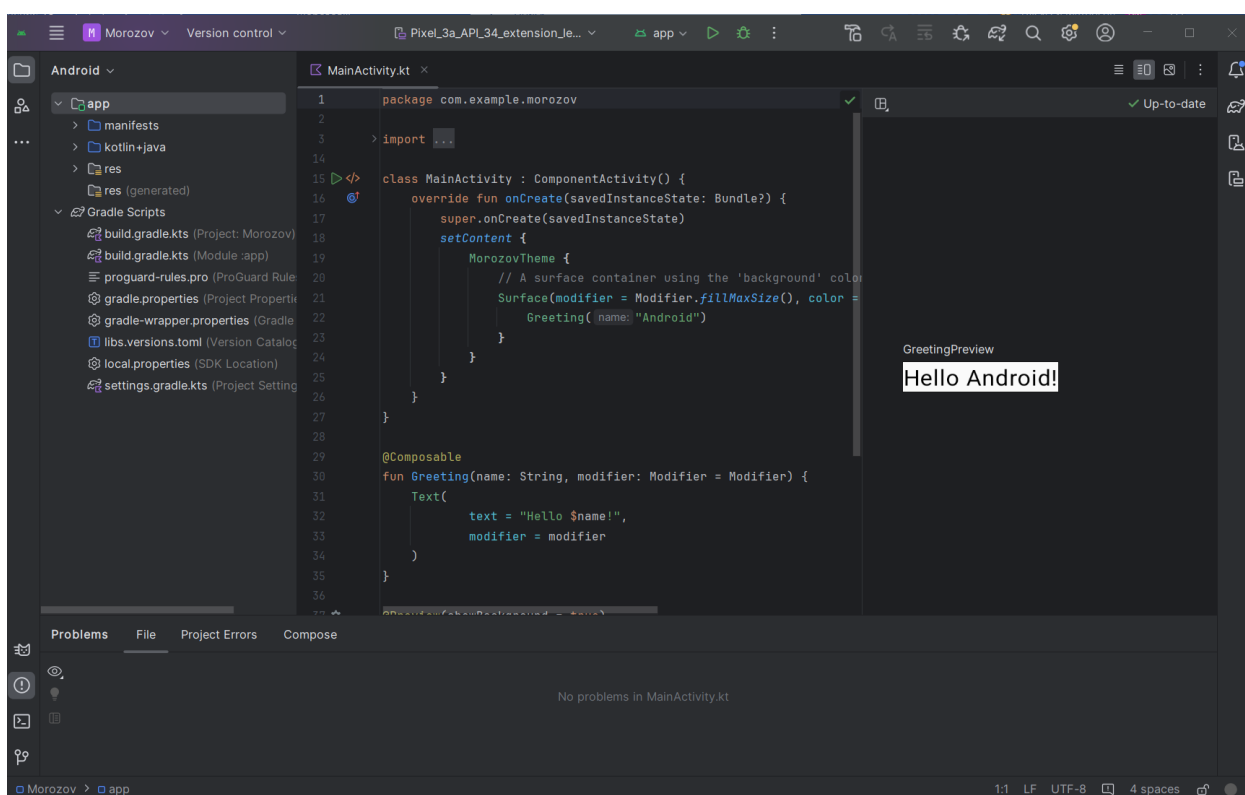


Рисунок 13 – Рабочее пространство Android Studio

Слева расположена древовидная структура проекта. В структуре проекта указаны используемые файлы, активные каталоги и библиотеки. В центральной части экрана расположено окно редактора кода [26]. Это основная область работы, в которой редактируется исходный код проекта [19]. Справа располагается окно превью, где в реальном времени отображается

визуальная интерпретация текущего кода. В нижней части экрана расположено окно отладки, в котором отслеживаются проблемы при сборке билда или отладке кода. Также Android Studio поддерживает запуск приложения на виртуальном эмуляторе для отладки и тестирования приложения в реальном времени [29].

Для своей работы мобильное приложение использует алгоритм аутентификации пользователя на базе OAuth2.0. OAuth 2.0 – это протокол аутентификации и авторизации, который позволяет приложениям получать доступ к данным пользователей на сторонних веб-сервисах без необходимости передачи им логина и пароля пользователя [28]. Алгоритм работает следующим образом:

- а) Пользователь инициирует процедуру аутентификации посредством нажатия на соответствующую кнопку в приложении.
- б) Приложение, которому требуется доступ к данным пользователя на стороннем веб-сервисе, инициирует процесс аутентификации и авторизации, направляя пользователя на страницу веб-сервиса.
- в) Веб-сервис перенаправляет пользователя на страницу входа.
- г) Пользователь вводит свои учетные данные на странице входа в веб-сервис. Веб-сервис проверяет эти учетные данные.
- д) Если пользователь успешно аутентифицирован, веб-сервис спрашивает пользователя о предоставлении разрешений на доступ к его данным (авторизация пользователя).
- е) Если разрешения подтверждены, веб-сервис предоставляет временный токен доступа (access token) приложению.
- ж) Приложение использует полученный токен доступа для запроса доступа к данным пользователя через API веб-сервиса.
- з) Веб-сервис проверяет токен доступа, чтобы убедиться, что он действителен и имеет права доступа, запрошенные приложением.

и) Если токен доступа действителен и имеет права доступа, указанные приложением, веб-сервис предоставляет доступ к данным пользователя.

Схематично принцип работы алгоритма представлен на рисунке 14.

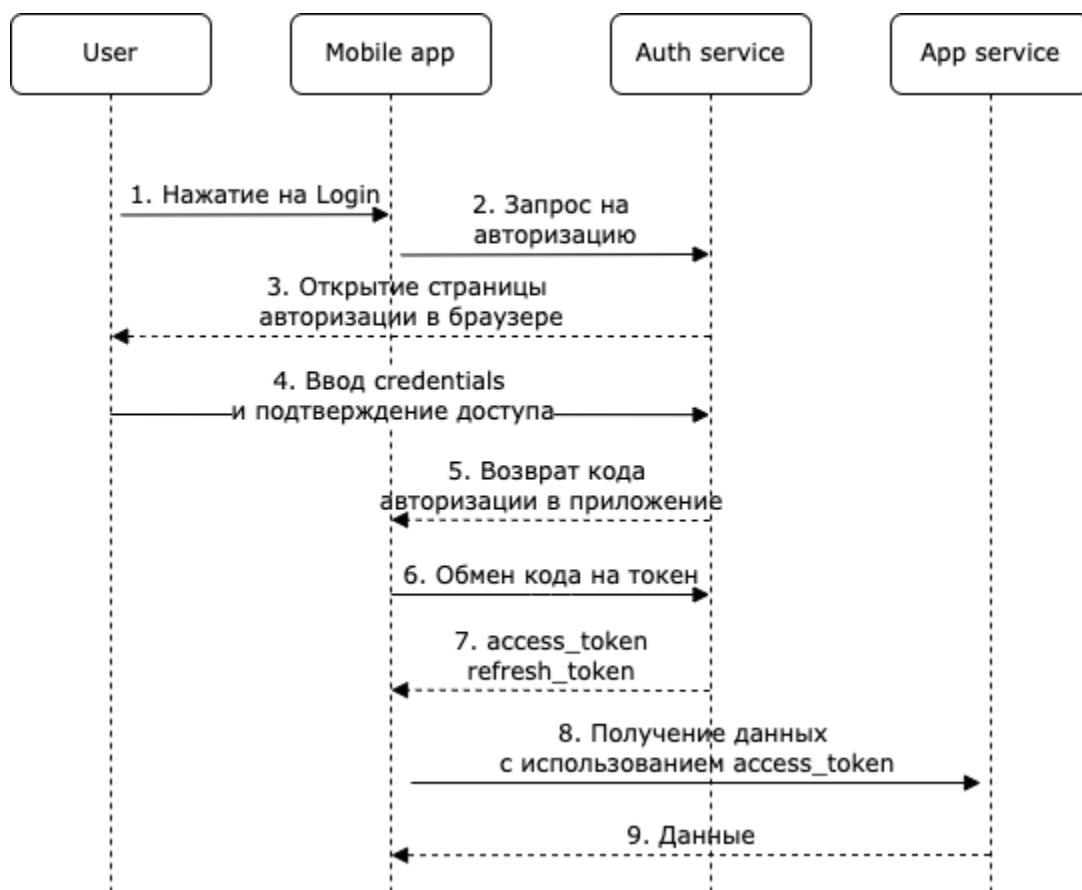


Рисунок 14 – Схема работы алгоритма авторизации OAuth 2.0

Форма аутентификации OAuth представляет собой веб-страницу, то есть требует интеграцию внешнего объекта внутри мобильного приложения [30]. Для отображения страницы можно использовать метод WebView [30], либо перенаправлять пользователя во внешний браузер для аутентификации [22], но этот вариант не слишком удобен. Поэтому для аутентификации в приложении был выбран вариант использования WebView. Чтобы воспользоваться возможностями WebView, необходимо настроить его в макете (рисунок 15).

```
<WebView
    android:id="@+id/webView"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Рисунок 15 – Настройка отображения WebView

Теперь вызов окна WebView будет отображаться корректно и занимать всё доступное приложению пространство. Данная настройка необходима для того, чтобы при изменениях размера окна приложения элементы формы оставались видимыми [27].

Чтобы обеспечить аутентификацию пользователя необходимо прописать функцию вызова окна WebView с указанной страницей OAuth (рисунок 16).

```
class OAuthWebViewActivity : AppCompatActivity() {

    companion object {
        const val AUTH_URL = "https://oauth.yandex.ru/authorize?response_type=code&client_id=ea135929105c4f"
    }

    private lateinit var webView: WebView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_oauth_webview)

        webView = findViewById(R.id.webView)

        // Настройка WebView
        webView.settings.javaScriptEnabled = true
        webView.webChromeClient = WebChromeClient()
        webView.webViewClient = object : WebViewClient() {
            override fun shouldOverrideUrlLoading(view: WebView?, request: WebResourceRequest?): Boolean {
                return false
            }
        }

        // Загрузка страницы авторизации
        loadAuthUrl()
    }
}
```

Рисунок 16 – Класс OAuthWebViewActivity, отвечающий за авторизацию

Такой подход позволяет упростить аутентификацию и авторизацию пользователя, фактически переключая процесс аутентификации на стороннего провайдера OAuth.

Главный экран приложения должен содержать элементы перехода к разделам о клиентах, задачах, текущих заказах и к продуктовому ассортименту. Создадим макет главного экрана при помощи xml (рисунок 17).

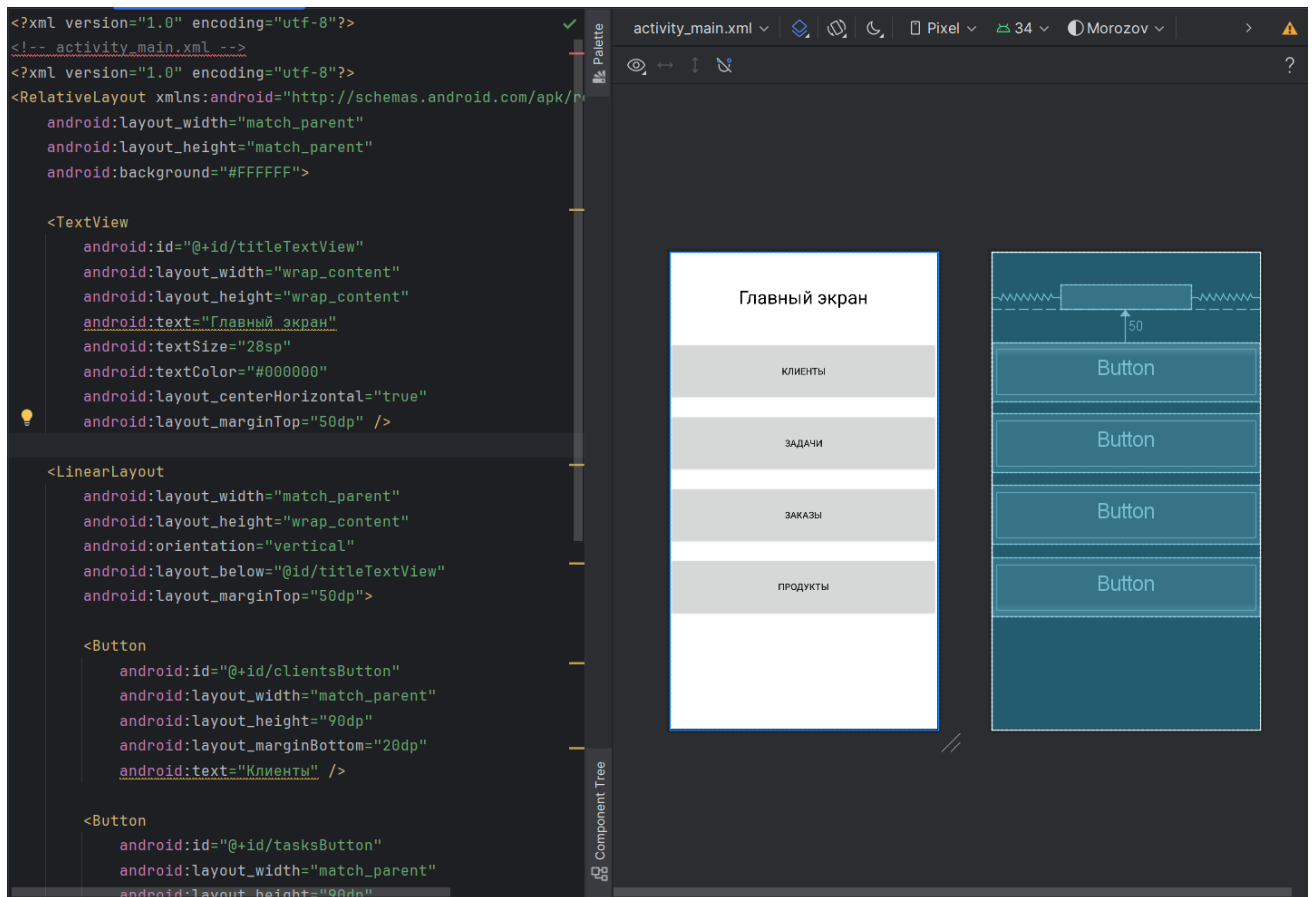


Рисунок 17 – Макет главного экрана и его исходный код

Для размещения элементов макета в вертикальном порядке используется RelativeLayout с базовыми атрибутами. Размещение горизонтальных элементов производится при помощи LinearLayout. Переходы

в нужные разделы приложения выполнены при помощи кнопок. Чтобы кнопки работали, необходимо добавить связанные с ними действия. Для этого используются обработчики нажатий. Чтобы кнопки корректно перенаправляли пользователя на нужный экран, добавим активности ClientsActivity, TasksActivity, OrdersActivity и ProductsActivity для соответствующих разделов.

Для корректного распознавания объекты можно найти при помощи их идентификаторов (рисунок 18).

```
val clientsButton = findViewById<Button>(R.id.clientsButton)
val tasksButton = findViewById<Button>(R.id.tasksButton)
val ordersButton = findViewById<Button>(R.id.ordersButton)
val productsButton = findViewById<Button>(R.id.productsButton)
```

Рисунок 18 – Использование поиска по идентификаторам

Метод findViewById используется в kotlin для указания объекта в виде строки и дальнейшего использования этого объекта в качестве строки [23]. Получив объекты, мы можем настроить обработчик нажатий для каждой кнопки при помощи OnClickListener, указывая необходимое событие при нажатии той или иной кнопки (рисунок 19).

```

clientsButton.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, ClientsActivity::class.java)
    startActivity(intent)
}

tasksButton.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, TasksActivity::class.java)
    startActivity(intent)
}

ordersButton.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, OrdersActivity::class.java)
    startActivity(intent)
}

productsButton.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, ProductsActivity::class.java)
    startActivity(intent)
}

```

Рисунок 19 – Назначение обработчиков событий

Таким образом, мы назначили старт активностей ClientsActivity, TasksActivity, OrdersActivity и ProductsActivity на соответствующие кнопки главного экрана. Перейдём к заполнению этих активностей.

Для начала необходимо создать макет списка клиентов. В списке должна отображаться основная информация – название организации и её адрес. Макет экрана клиентов и его код в xml представлен на рисунке 20.

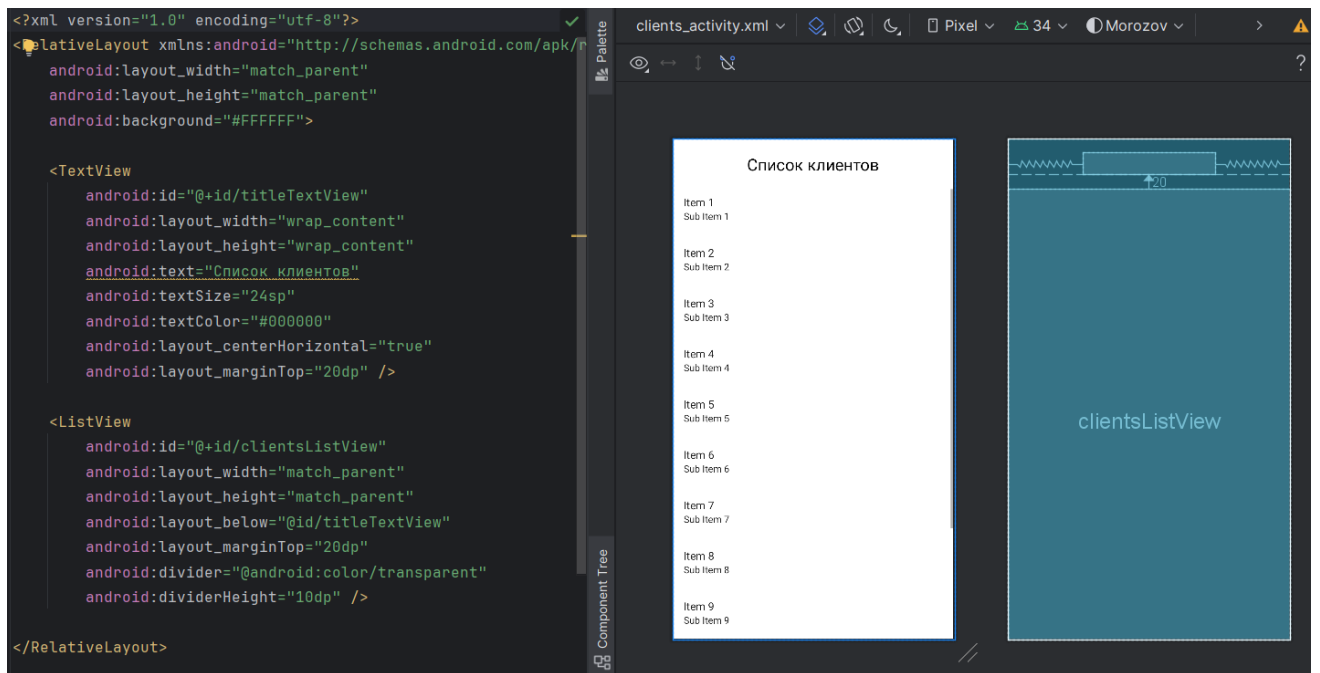


Рисунок 20 – Макет экрана клиентов

На данном макете выведен базовый список, позиции в котором мы будем менять при помощи стороннего класса в `ClientsActivity`. Чтобы клиенты корректно отображались, необходимо добавить массив клиентов и связать его адаптером с `ListView`. Для этого мы создадим соответствующие классы в `ClientsActivity` (рисунок 21).


```

class ClientsActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.clients_activity)

        val clients = listOf(
            Client( name: "000 Ромашка", address: "ул. Цветочная, 123"),
            Client( name: "000 Рога и Копыта", address: "пр. им. Крупного Рогатого Скота, 45"),
            Client( name: "000 Моя Оборона", address: "пер. Стелянного мира, 6")
        )

        val clientAdapter = ClientAdapter( context: this, clients)
        val clientsListView = findViewById<ListView>(R.id.clientsListView)
        clientsListView.adapter = clientAdapter
    }
}

class ClientAdapter(context: Context, clients: List<Client>) :
    ArrayAdapter<Client>(context, resource: 0, clients) {

    override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
        var itemView = convertView
        if (itemView == null) {
            itemView = LayoutInflater.from(context).inflate(R.layout.client_item, parent, attachToRoot: false)
        }

        val currentClient = getItem(position)

        val nameTextView = itemView!!.findViewById<TextView>(R.id.nameTextView)
        val addressTextView = itemView.findViewById<TextView>(R.id.addressTextView)

        currentClient?.let { it: Client
            nameTextView.text = it.name
            addressTextView.text = it.address
        }
    }
}

```

Рисунок 21 – Формирование списка клиентов и создание адаптера для
ListView

На данном этапе в целях тестирования клиенты жёстко прописаны внутри класса, но в дальнейшем осуществим импорт списка клиентов из внешних информационных систем, таких как CRM-система.

В таком формате мы получаем динамический объект, представленный на рисунке 22.

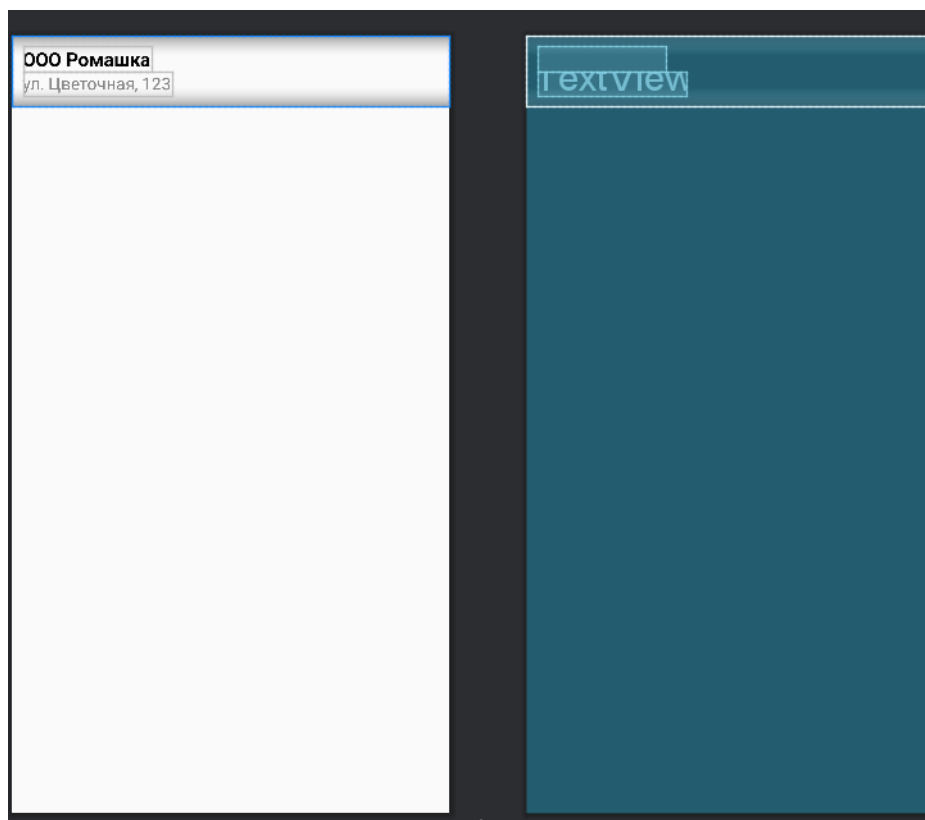


Рисунок 22 – Отображение динамического объекта `client_item`

Благодаря ранее написанному адаптеру мы заменяем стандартный список макета на такие динамические объекты, формируемые из списка `clients`.

Для отображения подробной информации о клиентах используется активность `ClientDetailsActivity` с макетом `client_details_activity.xml`. В ней указывается подробная информация – контактные данные, кнопка отображения истории заказов и кнопка построения маршрута, вызывающая API Google Maps (рисунок 23).

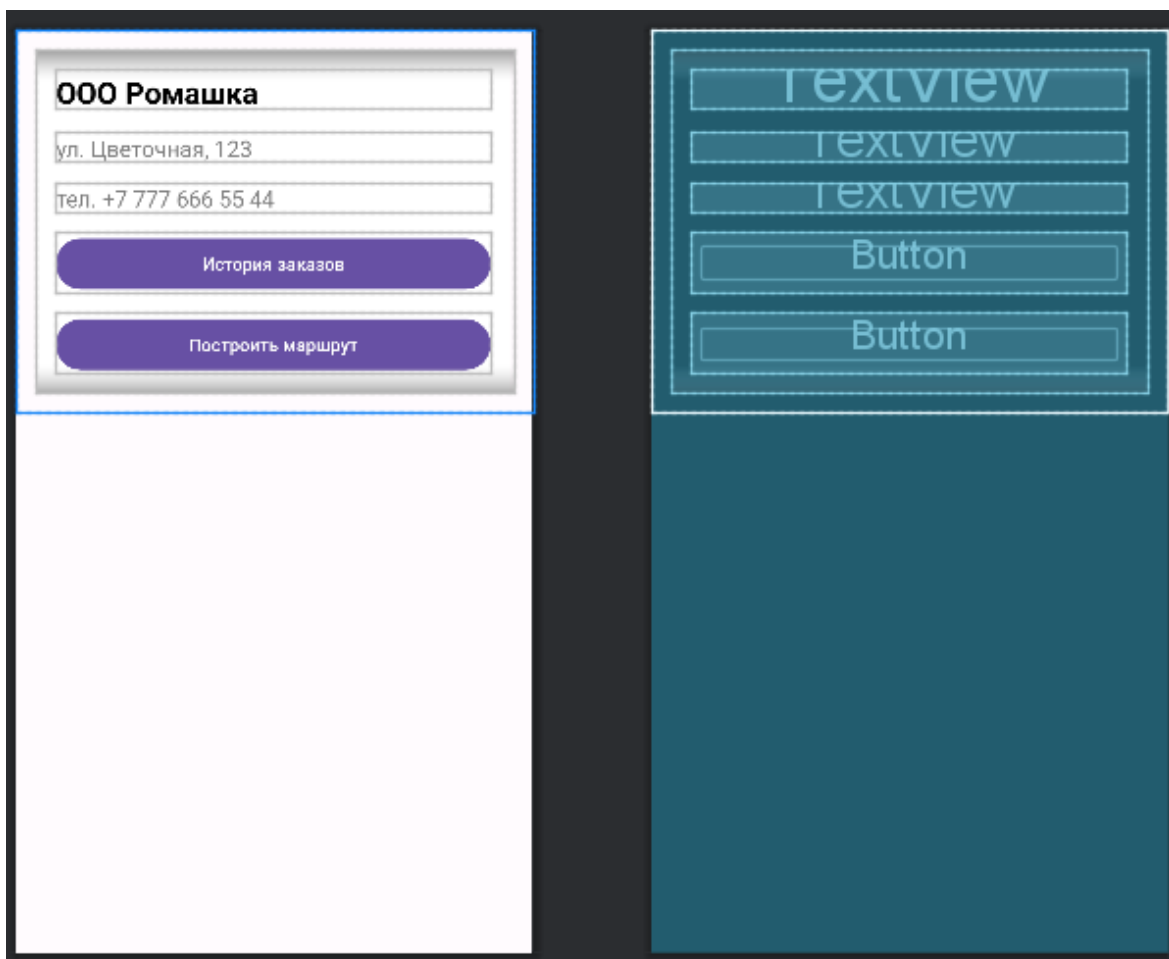


Рисунок 23 – Отображение подробной информации о клиенте

Для отображения поверх имеющейся информации мы используем метод `RelativeLayout` вместо стандартного `LinearLayout`. Такой подход позволяет не менять экран приложения, а вызвать дополнительное окно поверх предыдущего, отталкиваясь от первоначального расположения. Для вызова окна в активности необходимо задать метод `dialog.setContentview(R.layout.client_details_activity)`, и затем вызвать его при помощи `dialog.show()`.

Перейдём к созданию экрана задач торгового представителя. Для этого необходимо создать новый класс `TasksActivity.kt`, а также макет, используемый этим классом `activity_tasks.xml`. Код и отображение макета списка задач представлены на рисунке 24.

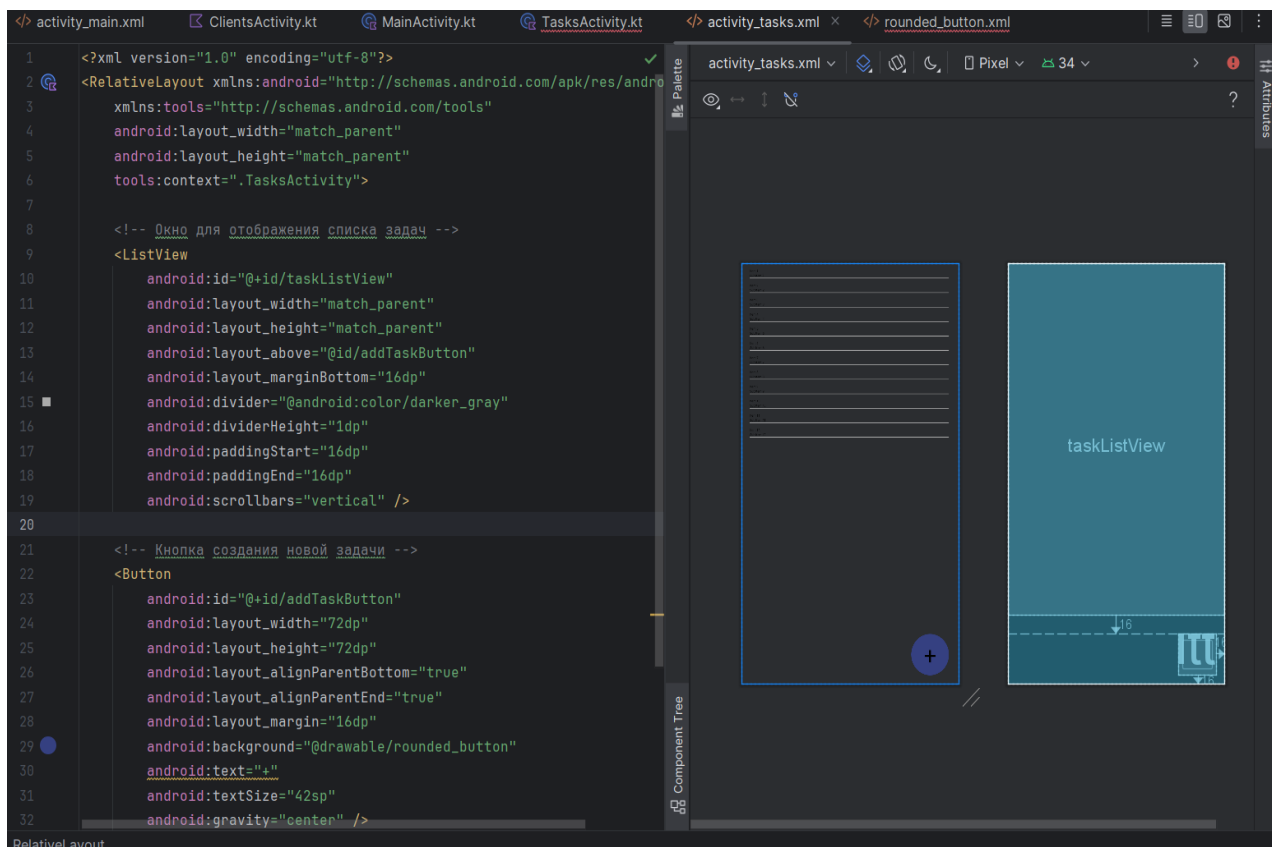


Рисунок 24 – Макет списка задач

При открытии экрана инициализируется загрузка задач из Google Calendar посредством API с использованием метода onCreate(). Инициализируются элементы интерфейса, в том числе кнопка добавления задачи addTaskButton и список отображения задач taskListView. Когда пользователь нажимает на кнопку addTaskButton, срабатывает слушатель событий, который запускает новую активность для создания новой задачи. После создания/изменения задачи или при запуске активности, загружается список текущих задач из Google Calendar. Полученный список задач отображается в списке taskListView. Для этого используется адаптер taskAdapter, который преобразует данные задач в представления, отображаемые в списках. Обновление задач реализовано через notifyDataSetChanged(), изменяющее представление в taskAdapter. Помимо основного функционала, такого как добавление и отображение задач, в TasksActivity также может быть реализована дополнительная

функциональность, такая как редактирование или удаление задач, фильтрация списка задач по различным критериям и т. д. При создании задач реализована возможность выбора связанного клиента посредством макета clientSpinner.

Продолжим создание приложения и перейдём к разделу заказов. Для заказов мы используем макет, схожий со списком клиентов (рисунок 25). Для динамической фильтрации заказов используется список ListView и адаптер для изменения содержимого списка в коде самой активности.

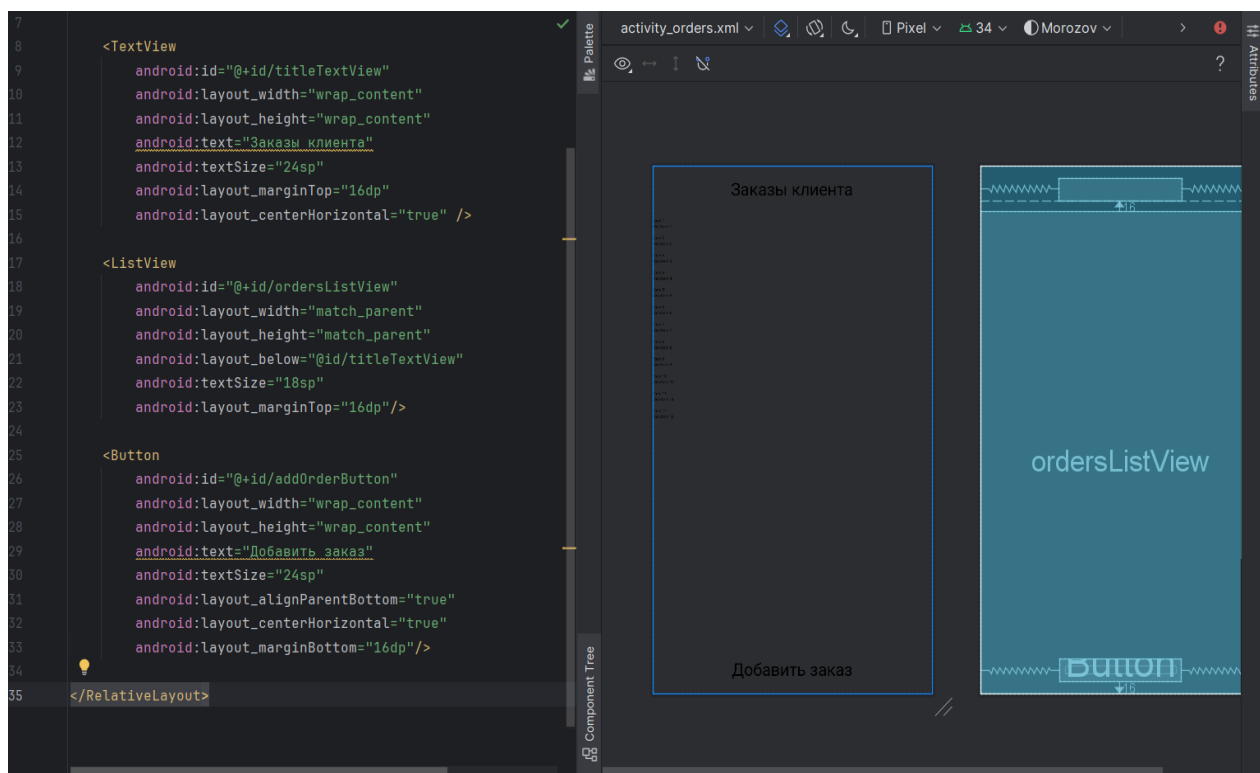


Рисунок 25 – Макет списка заказов и его исходный код

Логика реализована следующим образом: активность загружает макет activity_orders.xml, который содержит элементы ListView и Button. Для получения активного клиента используется параметр SELECTED_CLIENT, через который загружается список заказов из источника данных для выбранного клиента. Затем заказы сортируются и добавляются в ListView для отображения на экране. При нажатии на кнопку «Добавить заказ» происходит

редирект на экран добавления нового заказа AddOrderActivity. Исходный код логики активности представлен на рисунке 26.

```
private fun loadAndSortOrders(selectedClient: String) {
    orders.clear()
    val ordersForSelectedClient = getOrdersForClient(selectedClient)
    val sortedOrders = sortOrdersByClient(ordersForSelectedClient)
    orders.addAll(sortedOrders)
    orderAdapter.notifyDataSetChanged()
}

data class Order(
    val clientName: String,
    val orderDate: String,
    val orderNumber: String,
    val orderAmount: Double,
    val orderStatus: String
)

private fun getOrdersForClient(selectedClient: String): List<Order> {

return listOf(
    Order( clientName: "000 Ромашка", orderDate: "2024-05-24", orderNumber: "123", orderAmount: 10000.0, orderStatus: "Создан"),
    Order( clientName: "000 Рога и Копыта", orderDate: "2024-05-12", orderNumber: "124", orderAmount: 15000.0, orderStatus: "Завершен"),
    Order( clientName: "000 Моя Оборона", orderDate: "2024-05-17", orderNumber: "125", orderAmount: 20000.0, orderStatus: "Отгрузка")
)

private fun sortOrdersByClient(orders: List<Order>): List<Order> {
    return orders.sortedBy { it.clientName }
}
}
```

Рисунок 26 – Логика работы OrdersActivity

В дальнейшем предполагается интеграция заказов из сторонней CRM посредством API. На данный момент используется заглушка с фиксированными заказами. Для импорта заказов необходимо понимать, какие именно методы предоставляет API целевой CRM, а затем добавить обработчик событий в логику и сделать вывод на ListView макета.

Для удобства пользователя добавим возможность поиска заказов по любому полю. Представление поиска на макете изображено на рисунке 27.

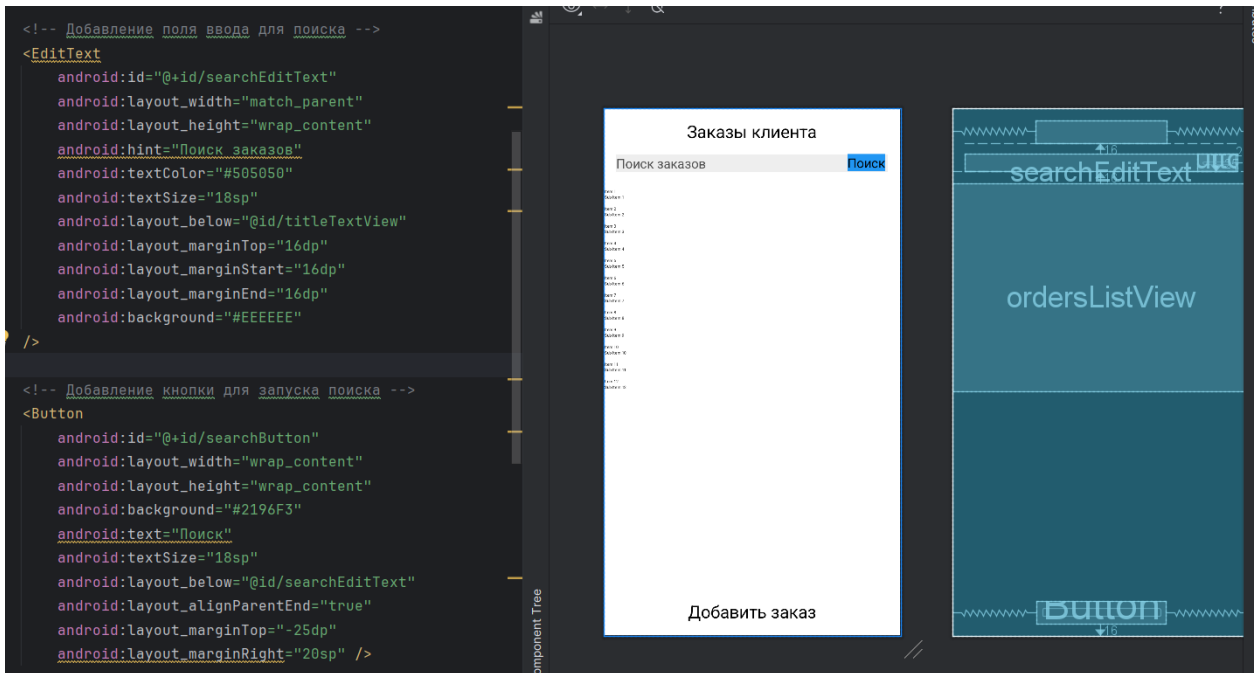


Рисунок 27 – Поиск по заказам

Теперь добавим соответствующую функцию в код самого приложения (рисунок 28).

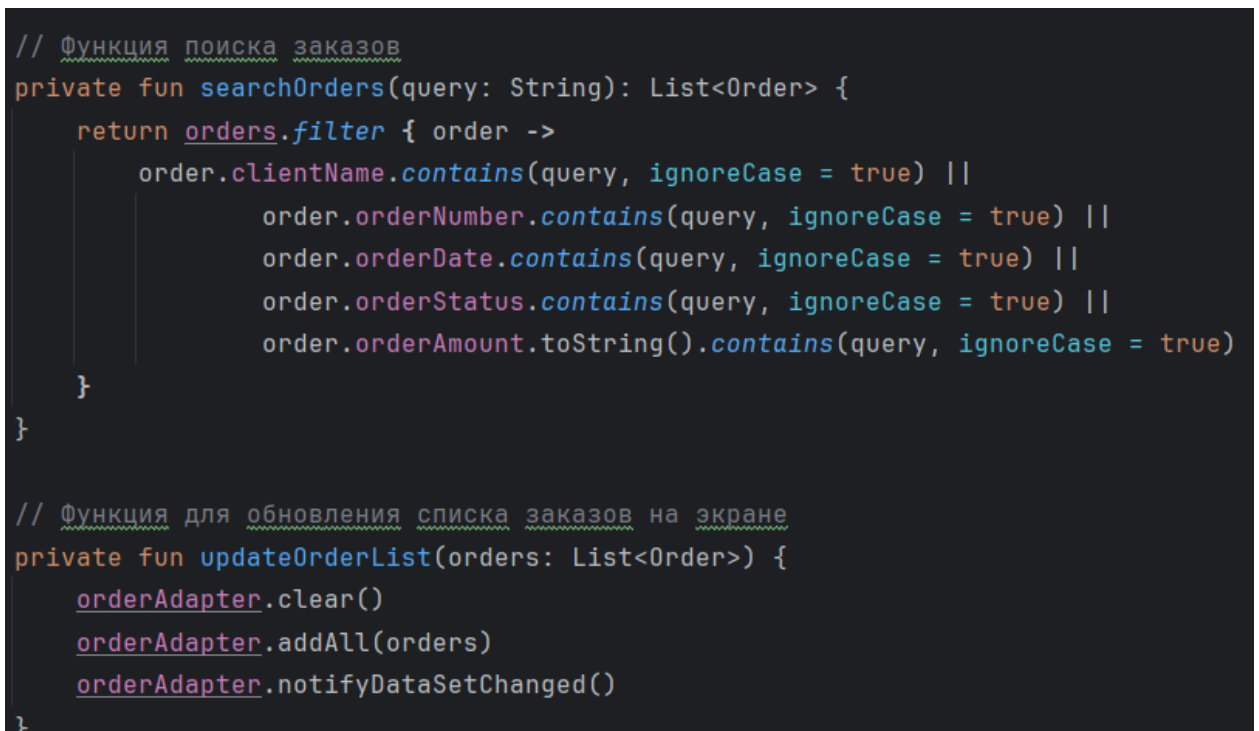


Рисунок 28 – Логика работы поиска по заказам

Для кнопки также создаётся обработчик событий, который запускает функцию поиска. Поиск проводится по всем колонкам заказа с игнорированием регистра, чтобы учесть ситуацию, когда пользователь вводит название компании или статус заказа с маленькой буквы.

Для формирования экрана продуктов требуется активная база 1С и тесная интеграция с ней, поэтому на данном этапе модуль реализовать в отрыве от внешнего сервера 1С не представляется возможным.

Таким образом, мобильное приложение имеет определённую функциональность и готово к проведению тестирования работоспособности.

3.2 Тестирование мобильного приложения

Тестирование работоспособности разработанного мобильного приложения осуществляется методом функционального тестирования. Функциональное тестирование подразумевает проверку основных функций приложения, корректное отображение интерфейса, тестирование вариантов использования и проверку стабильности работы приложения.

Сформируем программу тестирования приложения в таблице 2.

Таблица 2 – Функциональное тестирование мобильного приложения

№	Описание	Ожидаемый результат	Соответствие фактического результата ожидаемому
1	Валидация правильности входа пользователя с использованием правильных учетных данных	Пользователь успешно входит в систему с использованием правильных учетных данных	Частично
2	Проверка отображения сообщения об ошибке при вводе неверных учетных данных	При вводе неверных учетных данных пользователь видит сообщение об ошибке, указывающее на неверные данные	Соответствует
3	Просмотр списка клиентов	Пользователь видит список клиентов, доступных в приложении	Соответствует

Продолжение таблицы 2

№	Описание	Ожидаемый результат	Соответствие фактического результата ожидаемому
4	Добавление нового клиента	Пользователь успешно добавляет нового клиента, и этот клиент отображается в списке	Соответствует
5	Редактирование существующего клиента	Пользователь успешно редактирует данные существующего клиента, и изменения отображаются в списке	Соответствует
6	Удаление существующего клиента	Пользователь успешно удаляет существующего клиента, и этот клиент больше не отображается в списке	Соответствует
7	Просмотр списка задач	Пользователь видит список задач, доступных в приложении	Соответствует
8	Добавление новой задачи	Пользователь успешно добавляет новую задачу, и эта задача отображается в списке	Частично
9	Изменение существующей задачи	Пользователь успешно изменяет данные существующей задачи, и изменения отображаются в списке	Частично
10	Удаление задачи	Пользователь успешно удаляет задачу, и она больше не отображается в списке	Частично
11	Просмотр списка заказов для выбранного клиента	Пользователь видит список заказов для выбранного клиента	Соответствует
12	Добавление нового заказа	Пользователь успешно добавляет новый заказ для выбранного клиента, и этот заказ отображается в списке заказов для клиента	Соответствует
13	Редактирование существующего заказа	Пользователь успешно редактирует данные существующего заказа, и изменения отображаются в списке заказов для клиента	Не соответствует
14	Удаление заказа	Пользователь успешно удаляет заказ, и он больше не отображается в списке заказов для клиента	Не соответствует

Продолжение таблицы 2

№	Описание	Ожидаемый результат	Соответствие фактического результата ожидаемому
15	Поиск заказов по различным критериям (например, по дате или статусу)	Пользователь может выполнить поиск заказов по заданным критериям, и результаты соответствуют запросу	Соответствует
16	Проверка работы сортировки заказов по клиенту	Пользователь может отсортировать заказы по клиенту, и они отображаются в правильном порядке	Соответствует
17	Проверка отображения элементов интерфейса на устройствах с разной диагональю экранов	Элементы интерфейса корректно отображаются на устройствах с различными размерами экранов	Соответствует
18	Проверка навигации между экранами приложения	Пользователь может легко перемещаться между различными экранами приложения с помощью навигационных элементов	Соответствует
19	Проверка взаимодействия с элементами интерфейса, таких как кнопки, поля ввода	Пользователь может успешно взаимодействовать с элементами интерфейса, такими как кнопки и поля ввода, и они выполняют свои функции	Соответствует
20	Проверка безопасности передачи данных между приложением и сервером	Данные передаются между приложением и сервером безопасно и без утечек информации	Частично
21	Проверка совместимости приложения с различными версиями операционных систем и устройствами	Приложение корректно работает на различных версиях операционных систем и устройствах	Соответствует
22	Проверка производительности приложения, включая скорость загрузки и использование ресурсов устройства	Приложение работает быстро и эффективно, не перегружая ресурсы устройства	Соответствует

Из 22 пунктов тестирования полностью пройдены 15 пунктов. Валидация входа пока доступна в частичном представлении и работает только с тестовой учетной записью, без доступа по OAuth. Добавление новой задачи реализовано посредством вызова форму Google Calendar и не полностью

отражает привязку к клиенту. Изменение задачи также не учитывает контекст, но пользователь может редактировать задачу в календаре. Удаление задач производится посредством API Google. Также частично соответствует проверка безопасности передачи данных между приложением и сервером, поскольку не все сторонние серверы ещё подключены к приложению через API.

Не соответствуют ожиданиям функции редактирования существующего заказа и удаление, поскольку при вызове функций были выявлены ошибки работы приложения. В целом, приложение является функциональным и в дальнейшем может быть доработано до полноценного приложения, используемого в работе торгового представителя.

Выводы по главе 3.

В третьей главе бакалаврской работы были реализованы ранее спроектированные функции мобильного приложения и приведены в действующий вид.

Аутентификация и авторизация в мобильном приложении были реализована через OAuth 2.0 для интеграции со сторонними сервисами компании.

Был реализован главный экран приложения, разделы «Клиенты», «Задачи» и «Заказы». Каждый раздел позволяет добавлять новые записи, редактировать и удалять существующие, а также имеет сквозную связь между сущностями.

В ходе тестирования мобильного приложения была сформирована таблица функционального тестирования, и из 22 пунктов в ней удачно пройдены 15.

Заключение

В ходе выполнения бакалаврской работы был проведен обширный анализ и тщательное теоретическое исследование аспектов работы торгового представителя, сформированы требования к приложению, спроектирована его логика и выполнена реализация функционального мобильного приложения.

В первой главе бакалаврской работы был проведен анализ деятельности ООО «Вега-продукт» и работы торгового представителя. Была изучена структура организации и выявлены аспекты работы торгового представителя для будущего исследования.

При концептуальном моделировании и декомпозиции процессов работы торгового представителя были выявлены процессы, которые можно оптимизировать с помощью мобильного приложения. Декомпозиция процессов проводилась с использованием нотации IDEF0 для подробного описания рабочих процессов организации.

В результате был составлен список процессов для последующего логического моделирования приложения.

Во второй главе исследования создано логическое моделирование мобильного приложения для ОС Android. Определены требования к функциональности приложения по модели FURPS+ и разработаны соответствующие диаграммы использования и последовательности процессов. Выбрана среда разработки Android Studio после сравнительного анализа с IntelliJ IDEA и Eclipse.

В третьей главе работы были реализованы функциональные возможности мобильного приложения и представлены в рабочем состоянии.

Механизмы аутентификации и авторизации были осуществлены с применением протокола OAuth 2.0 для взаимодействия с корпоративными сервисами.

Разработан основной экран приложения, содержащий разделы "Клиенты", "Задачи" и "Заказы", обеспечивающие управление записями.

Проведено тестирование, в результате которого из 22 пунктов функционального тестирования успешно пройдены 15.

В ходе работы были применены методы анализа для выявления основных процессов работы торгового агента и выделения задач, решаемых при помощи мобильного приложения; методы моделирования для составления логических моделей разрабатываемого приложения; методы сравнения для выбора оптимальных инструментов при реализации приложения.

Задачи, поставленные в ходе работы, были выполнены в полном объёме. Мобильное приложение было спроектировано с учетом предъявляемых требований и может при дальнейшей доработке использоваться в работе торгового представителя. Таким образом, цель бакалаврской работы была достигнута.

Список используемой литературы

1. Аксенов К. В. Обзор современных средств для разработки мобильных приложений /К.В. Аксенов // Новые информационные технологии в автоматизированных системах. 2014. №17. С. 10-11.
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.
3. Басовский Л. Е. Комплексный экономический анализ хозяйственной деятельности : учеб. пособие / Л. Е. Басовский, Е. Н. Басовская. – Москва : ИНФРА-М, 2021. – 336 с.
4. Болхудере Е. И. Сравнение систем разработки мобильных приложений //XXIII Всероссийская студенческая научнопрактическая конференция Нижневартковского государственного университета. – 2021. – С. 91-97.
5. Вигерс К. Разработка требований к программному обеспечению.3-изд., дополнительное / К. Вигерс, Д. Битти М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014. 736 с.
6. Дейтел П. Android для разработчиков. – 3-е изд. / П. Дейтел, Х. Дейтел, А. Уолд. – Санкт-Петербург. : Питер, 2016. – 512 с.
7. Диго С.М. Проектирование и использование баз данных : Учебник / С.М. Диго, А.М. Вендеров. – М.: Финансы и статистика. 2015 г. – 592 с. – ISBN 5-279-02571-2.
8. Долганова, О. И. Моделирование бизнес-процессов : учебник и практикум для академического бакалавриата / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова ; под редакцией О. И. Долгановой. — Москва : Издательство Юрайт, 2019. — 289 с.
9. Дорохова А. М., Шацкий В. А., Картечина Н. В. Создание логической и физической модели базы данных //Наука и Образование. – 2020. – Т. 3. – №. 4. – С. 7.

10. Емельянова Н.З. Проектирование информационных систем: учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. М.: Форум, 2014. 432с
11. Ефимов Е.Н. Информационные системы в экономике. / Е.Н. Ефимов, С.М. Патрушина, Л.Ф. Панферова, Л.И. Хашиева. – Москва: ИКЦ «МарТ», 2004. – 352 с.
12. Жаров М. В. Основы информатики : учеб. пособие / М. В. Жаров, А. Р. Палтиевиц, А. В. Соколов. – Москва : Форум, 2017. – 512 с.
13. Заботина Н. Н. Проектирование информационных систем М.:ДРОФА, 2013. 336 с.
14. Иванова В. А. Путь аналитика. Практическое руководство IT-специалиста. 2-е изд. / В. А. Иванова, А. Ю. Перерва. – Санкт-Петербург. : Питер, 2018. – 304 с.
15. Колесов Ю. Б. Моделирование систем. Объектно-ориентированный подход: учебное пособие / Ю. Б. Колесов, Ю. Б. Сениченков. СПб.: БХВ-Петербург, 2012. 192 с.
16. 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
17. Новожилова А. Азбука клиента. Мобильные приложения:нативные, html5, // CMSMagazine. 2020. №1 (15). 97 с
18. Нужный, А. М. Разработка мобильных приложений : учебное пособие для СПО / А. М. Нужный, Н. И. Гребенникова, В. В. Сафронов. — Саратов : Профобразование, 2022. — 92 с.
19. Пирская, Л. В. Разработка мобильных приложений в среде Android 44 Studio : учебное пособие / Л. В. Пирская. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2019. — 123 с.
20. Роганов Е. А. Основы информатики и программирования : учеб. пособие / Е. А. Роганов. – 3-е изд. – Москва : Изд-во ИНТУИТ, 2021. – 390 с.

21. Смирнова Г. Н. Проектирование экономических информационных систем: учебное пособие / Г.Н. Смирнова, Ю.Ф. Тельнов. Международный консорциум "Электронный ун-т", Московский гос. ун-т экономики, статистики и информатики, Евразийский открытый ин-т. - Москва : МЭСИ, 2004. - Ч. 1. - 2004. - 222 с.
22. Утробина Елена Степановна, Кокорина Ирина Петровна Принципы разработки пользовательского интерфейса для мобильных картографических приложений // Интерэкспо Гео-Сибирь. 2019. №2.
23. Федотенко, М. А. Разработка мобильных приложений. Первые 45 шаги / М. А. Федотенко ; под редакцией В. В. Тарапаты. — Москва : Лаборатория знаний, 2019. — 336 с.
24. Чубенко М.Г., Смирнова Ю.А. Анализ мобильных приложений и мобильного рынка // Вестник науки. 2020. №6 (27). URL: <https://cyberleninka.ru/article/n/analiz-mobilnyh-prilozheniy-i-mobilnogorynka>
25. Эспозито, Д. Разработка современных веб-приложений: анализ предметных областей и технологий / Д. Эспозито. - М.: Вильямс И.Д., 2017. - 464 с.
26. Axelos, A. ITIL Practitioner Guidance. Stationery, 2017. – 231 с.
27. Dawn Griffiths. Head First Android Development: A Brain-Friendly Guide / Dawn Griffiths, David Griffiths - O'Reilly Media, 2015 - 734 p.
28. Helder Vasconcelos. Asynchronous Android Programming / Helder Vasconcelos - Packt Publishing, 2016. - 394 p.
29. Introduction to Android Architecture Components. [Электронный ресурс] / Режим доступа: URL: <https://medium.com/android-news/introductionto-android-architecture-components-22b8c84f0b9d> (дата обращения 21.03.2024)
30. Joshua Marinacci, Chris Adamson, Swing Hacks. Tips and Tools for Killer GUIs, 2017.