Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

Допускаю к защите

|  |  |
| --- | --- |
| Руководитель |  |
|  | подпись |
|  | З.А. Бахвалова |
|  | И.О. Фамилия |

|  |
| --- |
|  |
| Разработка прикладного программного обеспечения |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине

|  |
| --- |
| Технологии разработки программных комплексов |

|  |  |  |
| --- | --- | --- |
| 1.023.00.00 – ПЗ | | |
| обозначение документа | | |
| Выполнил студент |  | ИСТб-20-3 | |  |  | |  | Т.Е. Хомяков |
|  |  | шифр | |  | подпись | |  | И.О. Фамилия |
| Нормоконтроль |  |  | |  |  | |  | З.А. Бахвалова |
|  |  |  | |  | подпись | |  | И.О. Фамилия |
| Курсовая работа защищена с оценкой | | | | |  | | | |

Иркутск 2023 г.  
Иркутск 2023 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **По курсу** | Технологии разработки программных комплексов | | | | | | | | | | | | | | | | | | |
| **Студенту** | Хомякову Т.Е. | | | | | | | | | | | | | | | | | | |
|  | (фамилия, инициалы] | | | | | | | | | | | | | | | | | | |
| **Тема работы** | Разработка прикладного программного обеспечения | | | | | | | | | | | | | | | | | | |
| **Исходные данные** | | Учёт выполненных курсовых работ. | | | | | | | | | | | | | | | | |
| Проанализировать предметную область и реализовать систему в | | | | | | | | | | | | | | | | | | |
| соответствии с жизненным циклом ПО. | | | | | |  | | | | | | | | | | | | |
| **Рекомендуемая литература** | | | | | |  | | | | | | | | | | | | | |
| |  | | --- | | 1. СТО 0052020 СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА. Учебно | | методическая деятельность. Оформление курсовых проектов (работ) и | | выпускных квалификационных работ технических специальностей | | (<http://www.istu.edu/structure/57/2506/>); | | 1. Буч Г. Объектно-ориентированное проектирование с примерами применения: Пер. С англ. — М.: Конкорд, 1992519 с., ил | | 1. Курсовая работа «Разработка прикладного программного обеспечения» | | Руководство и методические указания для студентов направления | | 09.03.02 «Информационные системы и технологии». Бахвалова З.А. – | | Иркутск, Изд-во ИРНИТУ, 2018, 61 с. | | | | | | | | | | | | | | | | | | | | |
| Дата выдачи задания | | |  | | | 16 |  | февраля | | | |  | 2023 г. | | |
|  | | | | | | | | | | | | | | |
| Задание получил | | | |  | | | | |  | | Т.Е. Хомяков | | | | | |
|  | | | | подпись | | | | |  | | И.О. Фамилия | | | | | |
|  | | | | |  | |  | |  |  | | | |  | | | |
| Дата представления работы руководителю | | | | |  | |  | | мая | | |  | 2023г. | |
|  | | | | | | | | | | | | | | | |
| Руководитель курсовой работы | | | |  | | | |  | | | З.А. Бахвалова | | | | | |
|  | | | | подпись | | | |  | | | И.О. Фамилия | | | | | |

Оглавление

[**Введение** 4](#_Toc136385725)

[**1.** **Выработка системных требований** 5](#_Toc136385726)

[**1.1Описание предметной области** 5](#_Toc136385727)

[**1.2 Цель проекта** 5](#_Toc136385728)

[**1.3 Описание проблемы** 5](#_Toc136385729)

[**1.4** **Словарь предметной области** 6](#_Toc136385730)

[**1.5 Функциональность ПО** 7](#_Toc136385731)

[**2. Формальное описание требований к программному продукту** 8](#_Toc136385732)

[**2.1 Определение бизнес требований** 8](#_Toc136385733)

[**2.2 Определение требований пользователя** 10](#_Toc136385734)

[**2.3 Диаграмма интерфейсных классов** 15](#_Toc136385735)

[**2.4 Описание поведения разрабатываемой системы** 16](#_Toc136385736)

[**3. Проектирование программного продукта** 17](#_Toc136385737)

[**3.1Архитектура ПО** 17](#_Toc136385738)

[**3.1.1 Проектирование классов разрабатываемой системы** 17](#_Toc136385739)

[**3.1.3 Диаграмма классов** 18](#_Toc136385740)

[**3.1.4 Проектирование хранилища данных** 18](#_Toc136385741)

[**3.1.5** **Сценарии с учетом спроектированных классов данных и интерфейсных классов** 20](#_Toc136385742)

[**3.1.9 Проектирование макетов** 21](#_Toc136385743)

[**3.2 Программная реализация** 25](#_Toc136385744)

[**Заключение** 30](#_Toc136385745)

[**Список использованных источников** 31](#_Toc136385746)

**Введение**

Курсовая работа направлена на проектирование и создание многопользовательского приложения. В процессе выполнения курсовой работы студент отрабатывает технологический процесс разработки программного продукта, включающий в себя разработку технического задания, внешнее и внутреннее проектирование, кодирование, тестирование, разработку документации и развитие способности принимать обоснованные решения в ходе проектной деятельности и получать базовый опыт разработки программного обеспечения.

Для выполнения курсового проекта была выбрана тема «Система отслеживания статуса заказа». Цель проекта заключается в разработке программного обеспечения, которое поможет отслеживать статус заказа.

1. **Выработка системных требований**

**Описание предметной области**

Магазин цифровой техники - это коммерческая организация, где продаются товары, связанные с электроникой, компьютерами, мобильными устройствами и другой цифровой техникой.

В магазине цифровой техники представлен широкий ассортимент товаров, включая ноутбуки, персональные компьютеры, смартфоны, планшеты, телевизоры, камеры, звуковое оборудование, наушники, игровые приставки и многое другое. Также магазины цифровой техники могут предоставлять услуги по ремонту и обслуживанию техники, консультации по выбору устройств и другие дополнительные услуги.

Покупатели могут осуществлять покупку товаров как в магазинах, так и онлайн.

Для эффективного управления бизнесом и обеспечения удобства для покупателей магазины цифровой техники используют программные системы, которые автоматизируют процессы управления складом, учета продаж, рассылки рекламных материалов и другие задачи.

**1.2 Цель проекта**

Увеличить удовлетворенность клиентов и укрепить доверие к компании. Необходимо предоставлять клиентам достаточно информации о статусе и местоположении их заказов.

**1.3 Описание системы**

Система позволит клиентам проверять статус своих заказов на каждом этапе доставки:

1. Заказ оформлен

2. Заказ на сборке

3. Заказ передан курьеру

4. Доставлено

Программа будет иметь графический интерфейс пользователя, позволяющий ему отслеживать статус заказов и получать информацию о доставке. Для этого программа будет использовать базу данных, где хранится информация о заказах и их статусе. Программа будет иметь возможность автоматически отправлять уведомления клиентам о статусе их заказа, что поможет снизить количество запросов в поддержку.

* 1. **Словарь предметной области**

Таблица 1 – Словарь терминов предметной области

|  |  |  |
| --- | --- | --- |
| № | Термин | Назначение |
| 1. | Товар | Продукт, предлагаемый магазином цифровой техники, такой как компьютеры, телефоны, планшеты, ноутбуки, аксессуары и т.д. |
| 2. | Курьер | Человек или компания, ответственные за доставку товаров клиентам. |
| 3. | Статус заказа | Текущий этап выполнения заказа, например, обработка, упаковка, доставка, доставлено. |
| 4. | Клиент | Человек или организация, которые хотят купить товар в магазине цифровой техники. |
| 5. | Заказ | Запрос клиента на покупку определенного товара. |
| 6. | Трекинг-номер | Уникальный идентификатор, который позволяет клиентам отслеживать статус своего заказа. |
| 7. | Доставка | Процесс доставки товаров клиенту. |
| 8. | Упаковка | Материалы, используемые для защиты товаров во время транспортировки. |

Таблица 2 – Словарь предметной области по методу Аббота

|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Глагол** |
| 1 | Товар | Выбирать, искать |
| 2 | Курьер |  |
| 3 | Статус заказа |  |
| 4 | Клиент |  |
| 5 | Заказ | Оформлять, отслеживать |
| 6 | Трекинг-номер |  |
| 7 | Доставка |  |
| 8 | Упаковка |  |

Таблица 3 – Объектно-ориентированный словарь

|  |  |  |  |
| --- | --- | --- | --- |
| № | Класс (Сущность/Актер) | Свойство (Состояние) | Метод (Функция/Поведение) |
| 1 | Товар | Модель, картинка, артикул, цена, цвет, материал, наличие | Выбирать, искать |
| 2 | Курьер | Идентификатор курьера, имя, фамилия, контактные данные |  |
| 3 | Статус заказа | Идентификатор статуса, название | Выбирать, смотреть |
| 4 | Клиент | Идентификатор клиента, имя, фамилия, контактные данные |  |
| 5 | Заказ | Товар, Дата, Стоимость | Оформлять |
| 6 | Трекинг-номер | Уникальный идентификатор заказа |  |
| 7 | Доставка | ФИО покупателя, адрес | Оформлять |
| 8 | Упаковка | Материалы |  |

**2. Формальное описание требований к программному продукту**

## **2.1 Модель предметной области (IDEF0)**

Данная модель IDEF0 описывает процесс «Система отслеживания статуса заказа».

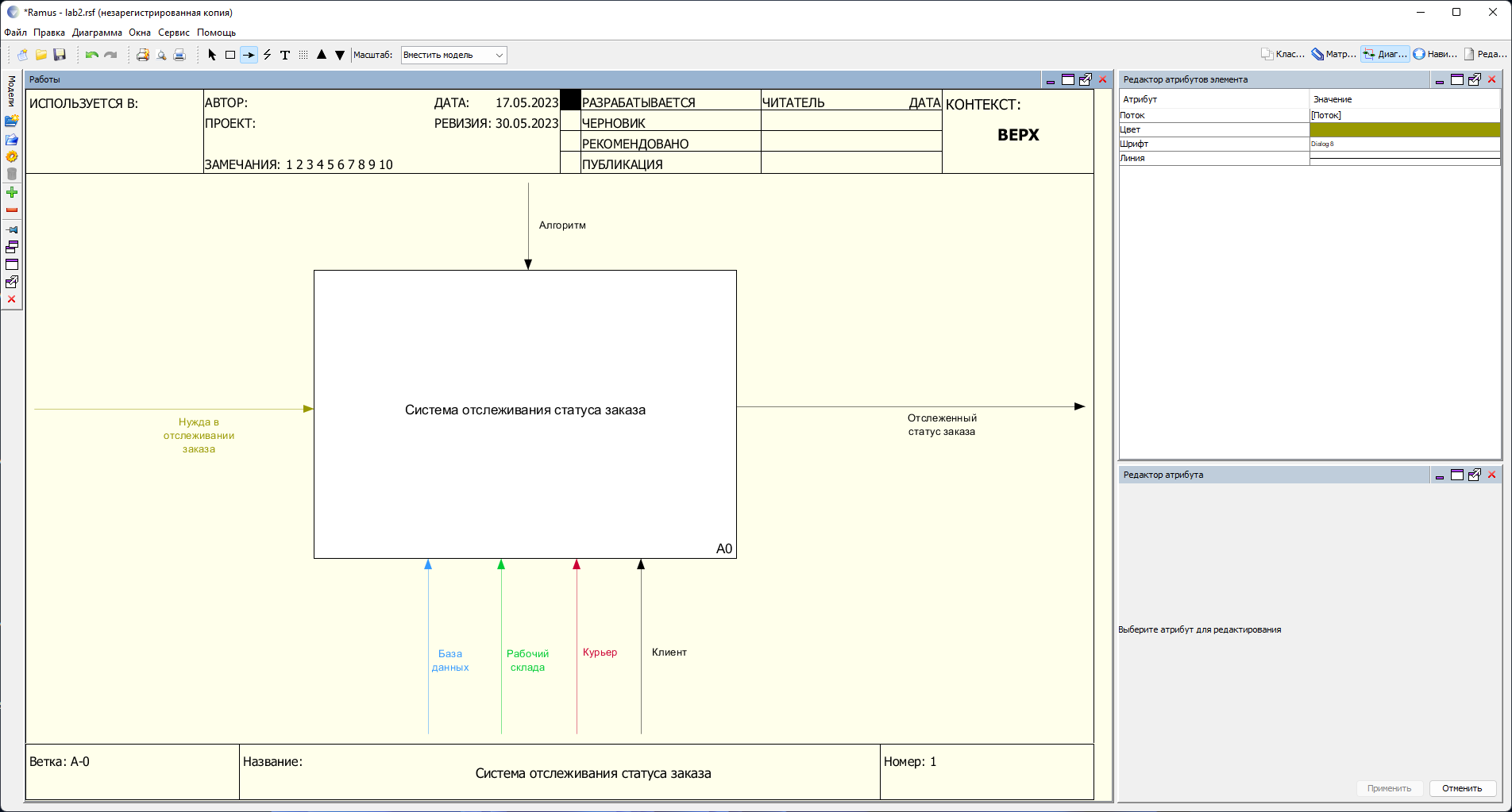


Рисунок 1 – Модель IDEF0. Уровень А0. (Система отслеживания статуса заказа)

Клиент вводит идентификатор заказа, который получает при оплате заказа.

База данных рейсов хранит информацию о заказах и клиентах. Сотрудник магазина создает заказ в базе данных. Курьер и Рабочий склада изменяют статус заказа.

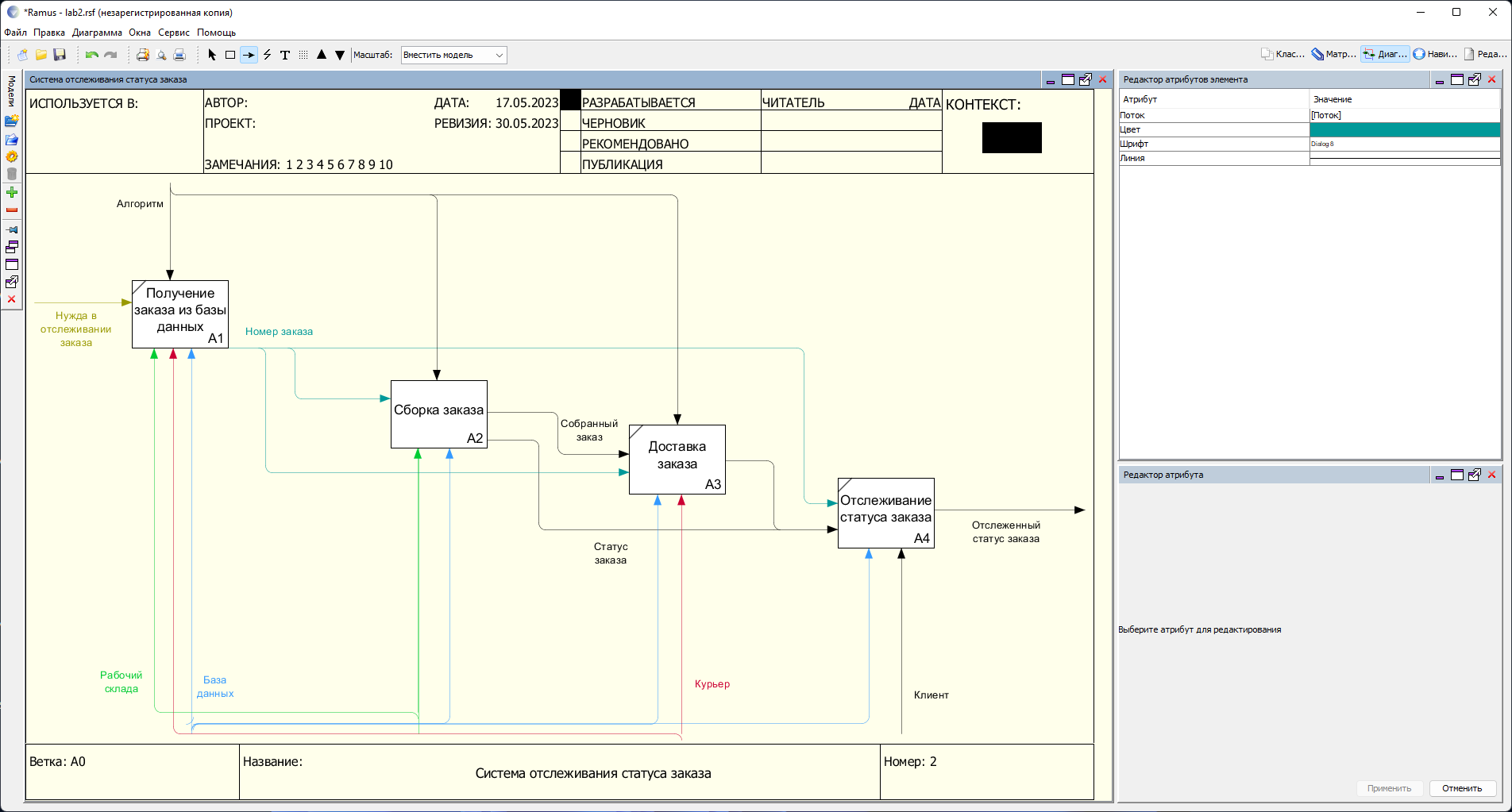


Рисунок 2 – Модель IDEF0. Уровень А1. (Система отслеживания статуса заказа)

Процесс А1 «Система отслеживания статуса заказа» имеет детализацию, представленную на рисунке 2.

В начале процесса заказ добавляется в базу данных, в процессе доставки статус заказа изменяется, в это время клиент может отслеживать статус своего заказа.

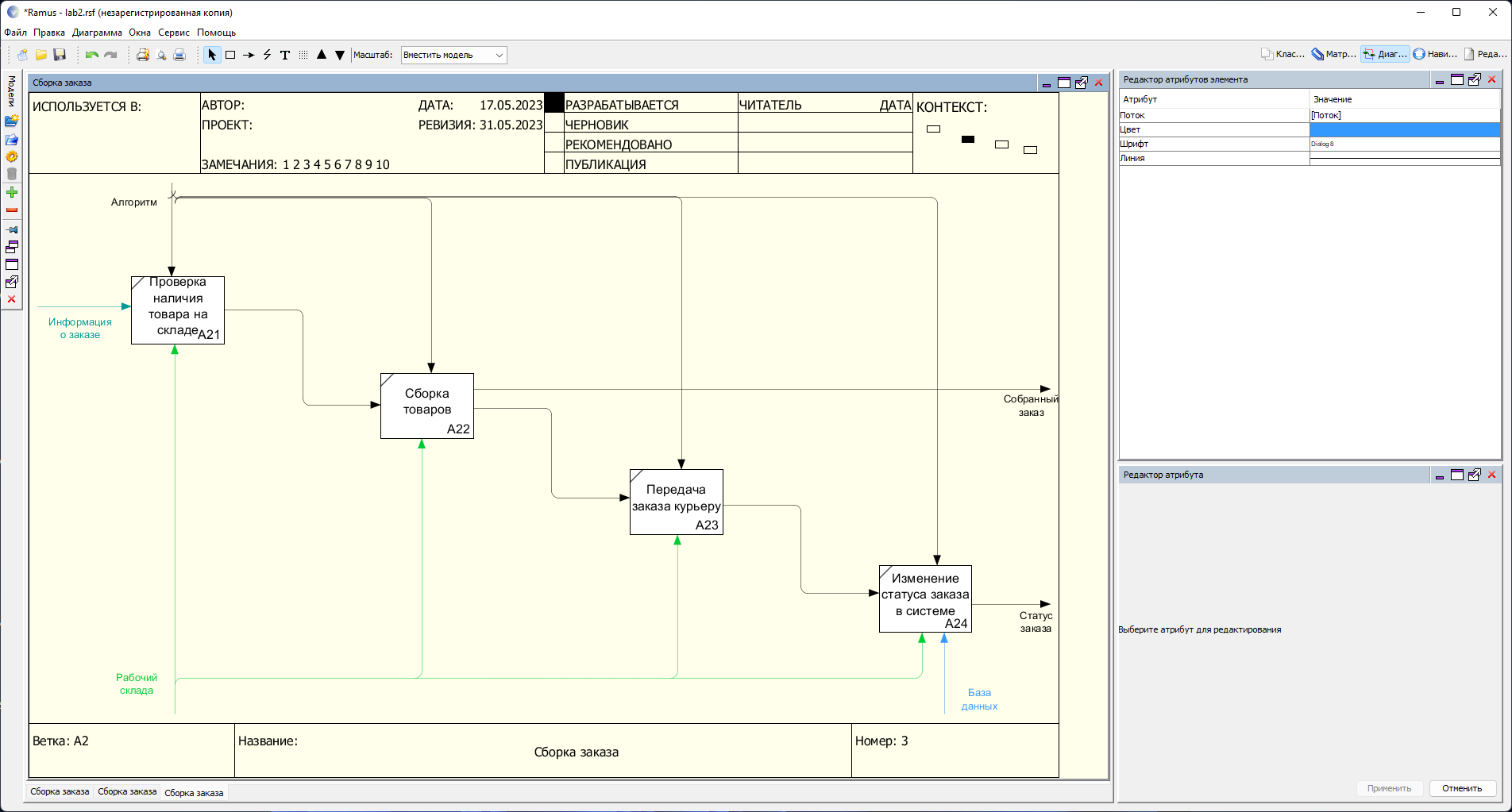


Рисунок 3 – Модель IDEF0. Уровень А2. (Сборка заказа)

Процесс А2 «Сборка заказа» имеет детализацию, представленную на рисунке 3.

**2.2 Определение требований пользователя**

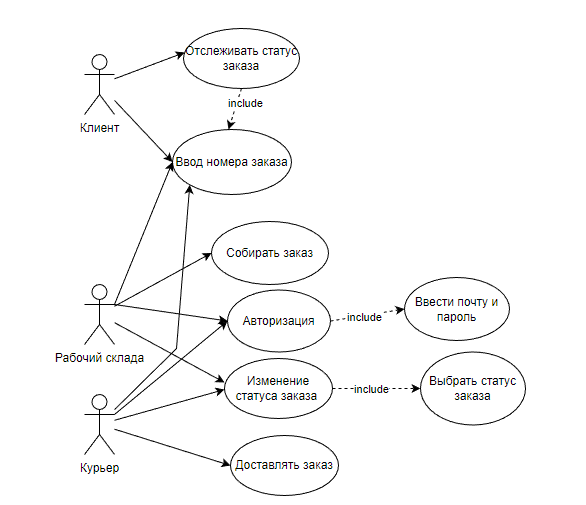


Рисунок 4 – Диаграмма вариантов использования.

Клиент, рабочий склада, курьер могут выполнять функции, которые отображены на рисунке 4.

### **Вариант использования «Отслеживание заказа».**

Спецификация варианта использования «Отслеживание заказа».

**Цель:** Отследить статус заказа.

**Активные субъекты:** Клиент.

**Краткое описание:** Активный субъект имеет возможность отследить статус заказа.

**Предусловия:** необходимо *«отследить статус заказа»*.

**Постусловия:** нет.

**Основной поток событий:**

1. Клиент вводит идентификатор заказа.
2. **Альтернативные потоки событий:** нет.

**Сценарий обработки ошибок:** нет.

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

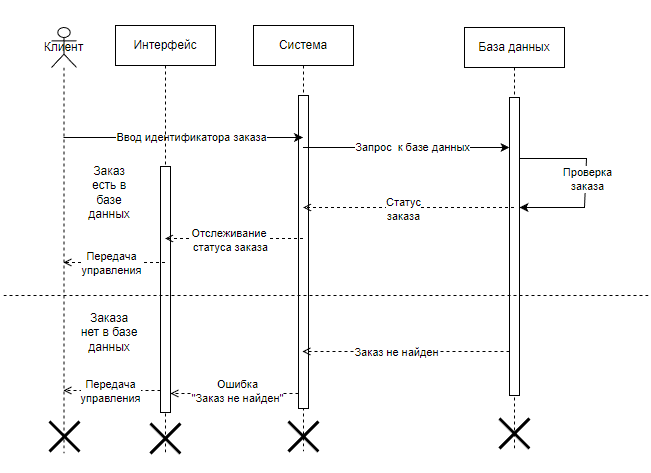


Рисунок 8 – Диаграмма последовательностей варианта использований

*«Отслеживание заказа»*.

### **5.2 Вариант использования «Изменение статуса заказа».**

Спецификация варианта использования «Изменение статуса заказа».

**Цель:** получить отсортированный список товаров.

**Активные субъекты:** Курьер, рабочий склада.

**Краткое описание:** Активный субъект имеет возможность изменять статус заказа.

**Предусловия:** необходимо *«Изменить статус заказа»*.

**Постусловия:** нет.

**Основной поток событий:**

1. Курьер и рабочий склада изменяет статус заказа

**Альтернативные потоки событий:** нет.

**Сценарий обработки ошибок:**

**Специальные требования:** нет.

**Дополнительные замечания:** нет.

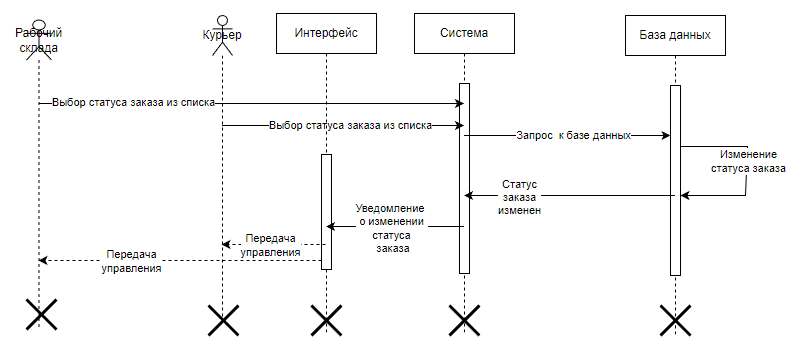


Рисунок 9 – Диаграмма последовательностей варианта использований *«Изменение статуса заказа»*.

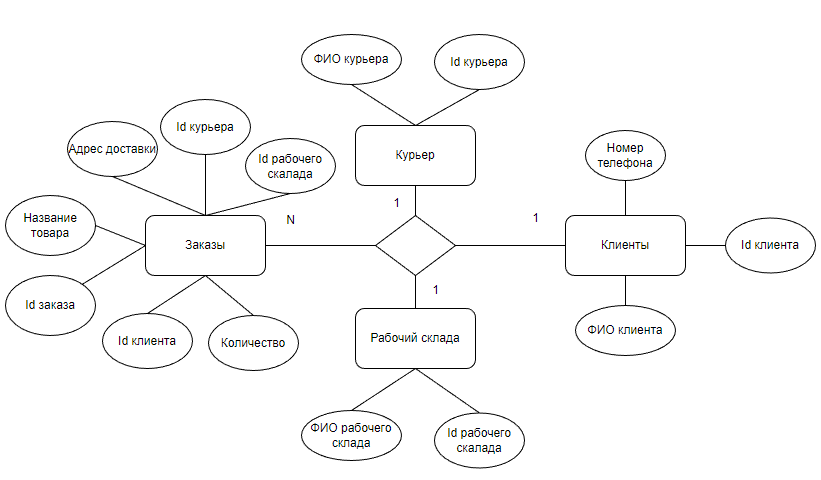


Рисунок 7 – ER-модель.

**2.3 Диаграмма интерфейсных классов**

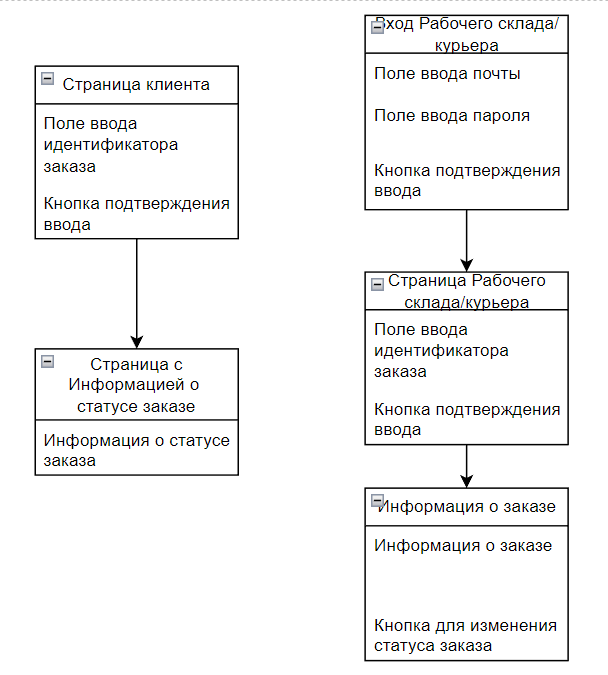


Рисунок 16 – Диаграмма интерфейсных классов

На рисунке 16 изображена диаграмма интерфейсных классов, где отображены главные составляющие интерфейса. Есть 5 основных страниц: Страница входа, Страница клиента, Информация о статусе заказа, Страница Рабочего склада/курьера, Информация о заказе.

**2.4 Описание поведения разрабатываемой системы**

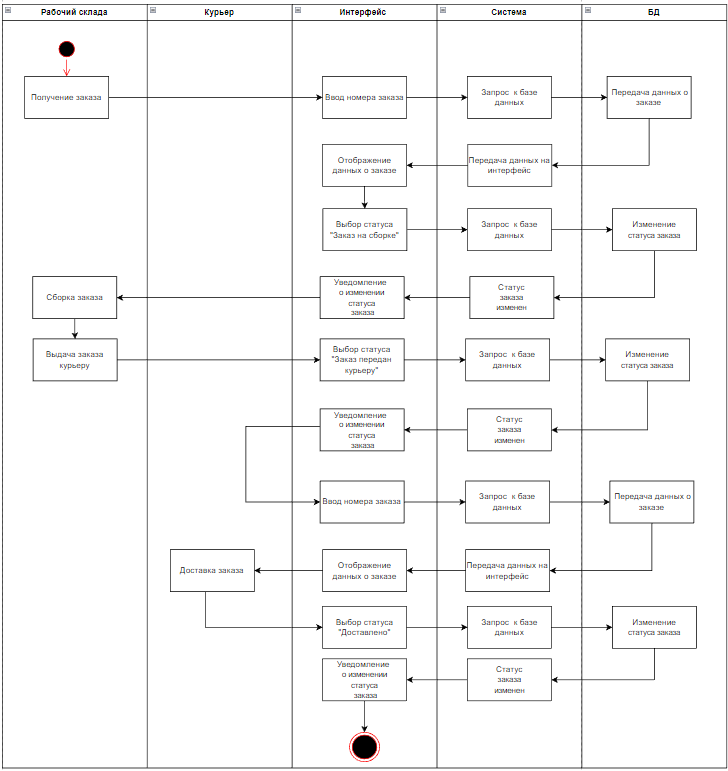


Рисунок 11 - Диаграмма активности. Изменение статуса заказа

# **3. Проектирование программного продукта**

**3.1Архитектура ПО**

**3.1.1 Проектирование классов разрабатываемой системы**

Для создания программы, будет использована среда разработки android studio, язык программирования Kotlin. Данная IDE позволяет разрабатывать мобильные приложения для системы Android. Кotlin - это современный, статически типизированный язык программирования, разработанный для платформы Java. Он обеспечивает множество преимуществ, делая его отличным выбором для написания приложений отслеживания статуса заказа. Kotlin предлагает интероперабельность с Java, что позволяет использовать уже существующий код и интегрировать Kotlin-модули в Java-проекты без проблем, так же мы использовали firebase — облачная СУБД класса NoSQL, позволяющая разработчикам приложений хранить данные и синхронизировать их между несколькими клиентами.

Архитектура, описанная в тексте, является клиент-серверной архитектурой. В данном случае программа, написанная на языке Kotlin и разработанная в среде Android studio, выступает в роли клиента, который взаимодействует с базой данных, созданной в Firebase.

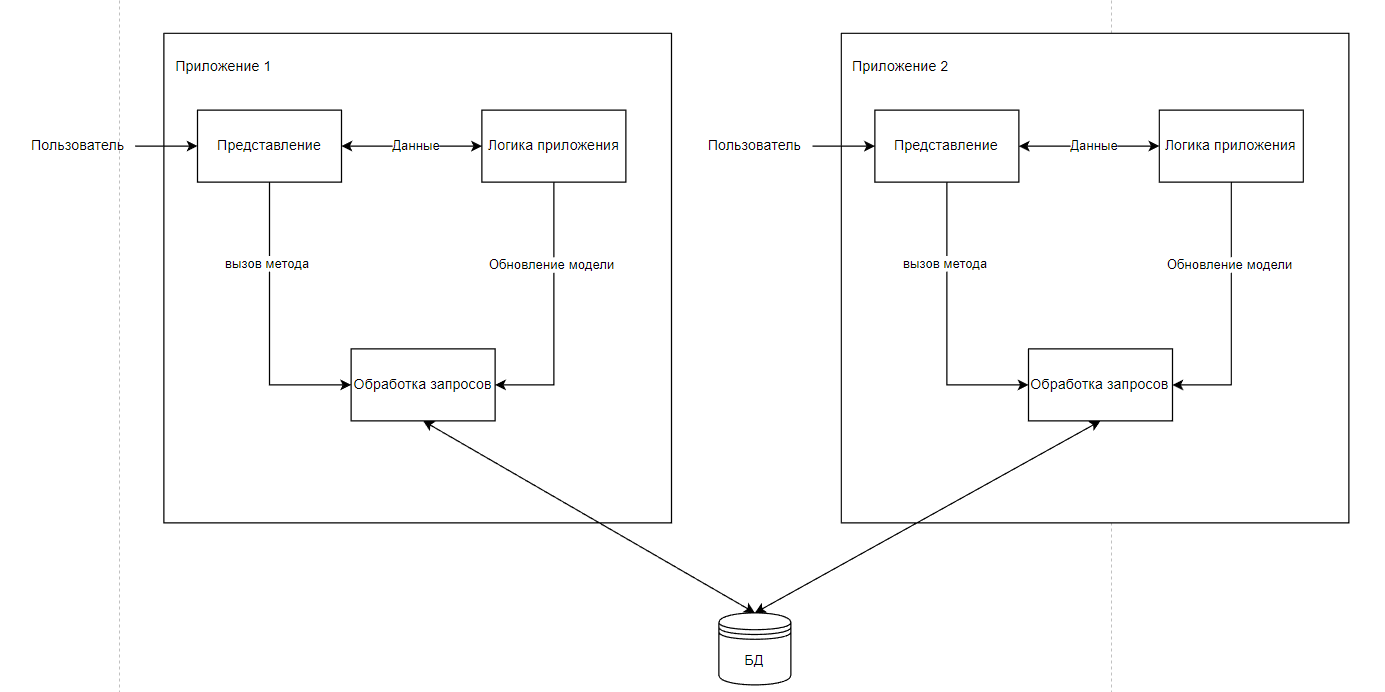
 

Рисунок 12 – Функциональная модель

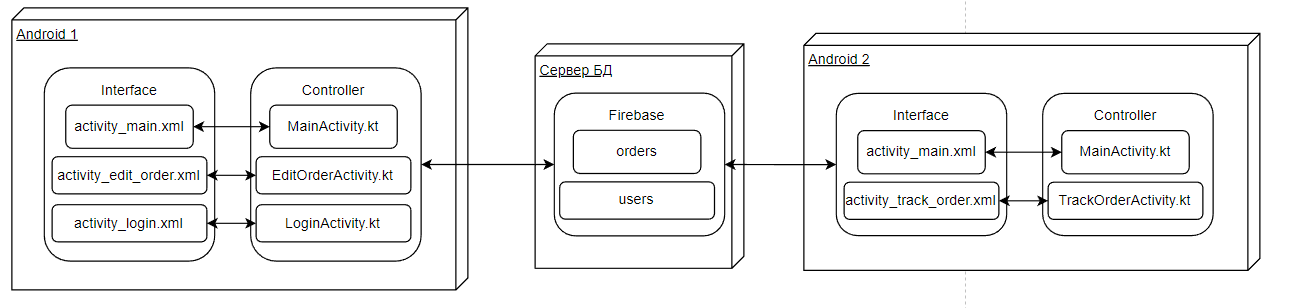


Рисунок 13 – Декомпозиция функциональной модели

На диаграмме развертывания представлены узлы устройств, которые требуются для реализации ПО, а также связи показывающие взаимодействия.

**3.1.3 Диаграмма классов**

На рисунке 15 предоставлена диаграмма классов. Диаграмма классов необходима для описания функциональности и поведения программы.

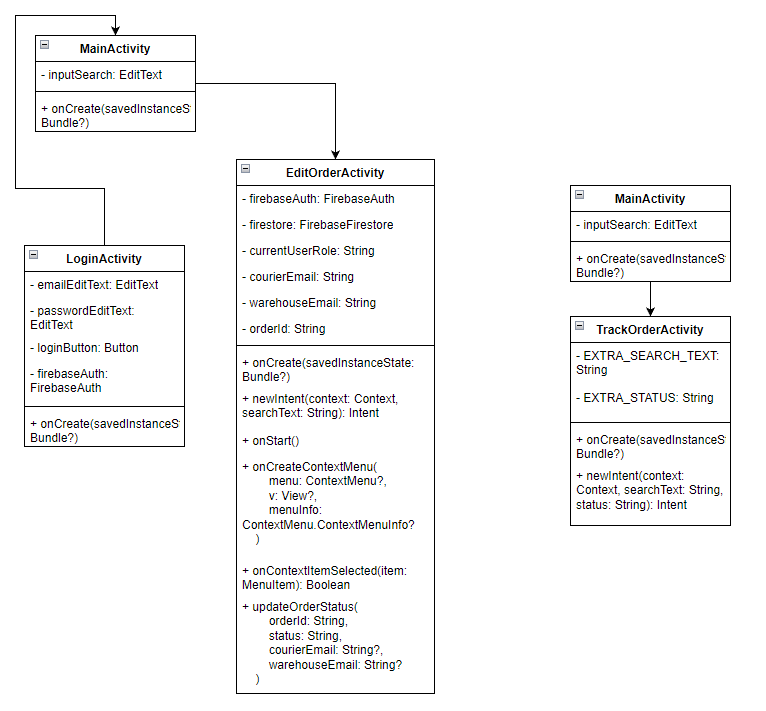


Рисунок 18 – Диаграмма классов приложения.

**3.1.4 Проектирование хранилища данных**

На рисунке 19 представлена логическая модель представления данных

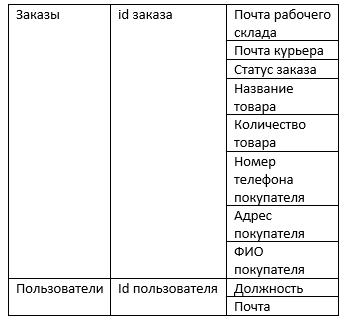


Рисунок 19 - Логическая модель представления данных

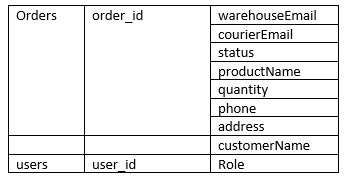


Рисунок 20 – Физическая схема БД

* + 1. **Сценарии с учетом спроектированных классов данных и интерфейсных классов**

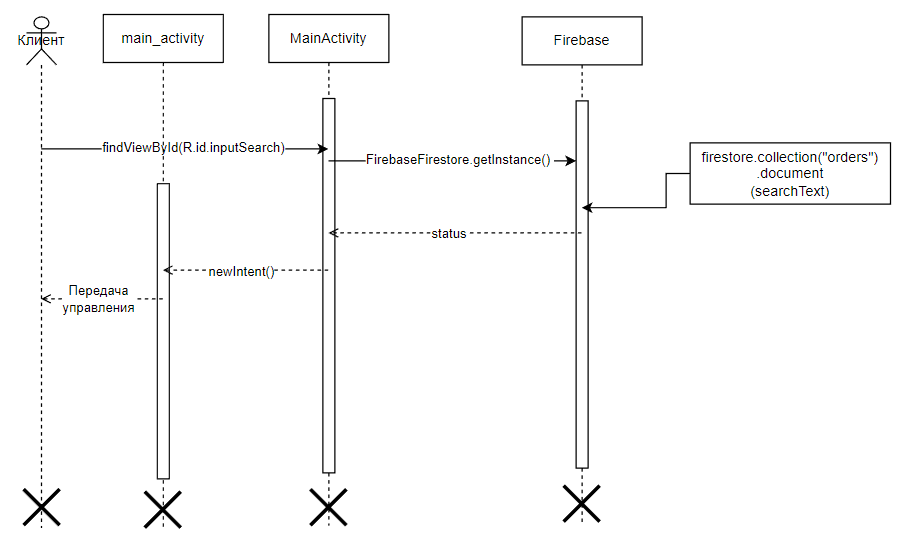


Рисунок 21 -Проектирование Use Case «Отслеживание статуса заказа»

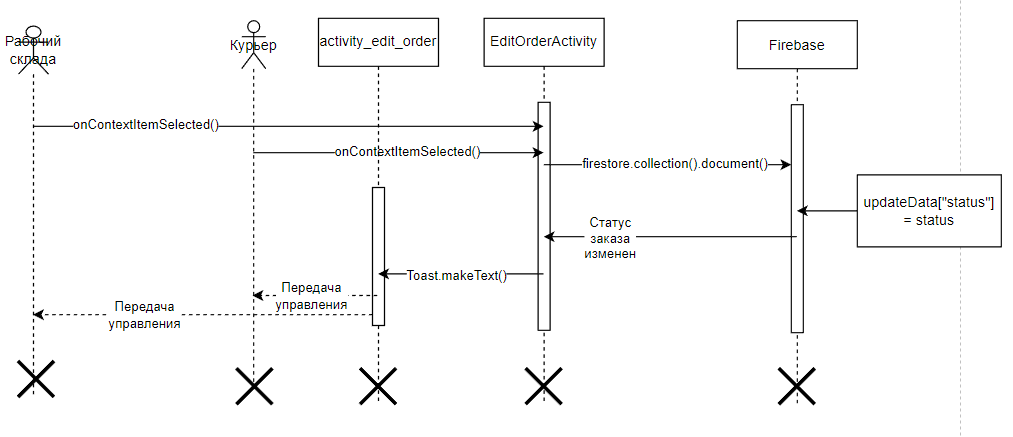


Рисунок 22 - Проектирование Use Case «Изменение статуса заказа»

**3.1.9 Проектирование макетов**

Для создания графического интерфейса приложения будем использовать XML layout.

На рисунке 18 изображён макет окна входа (activity\_login). Именно это окно видит пользователь, когда запускает приложение.

Окно содержит следующие основные компоненты: два поля ввода (emailEditText, passwordEditText) и кнопка «Войти»( loginButton).

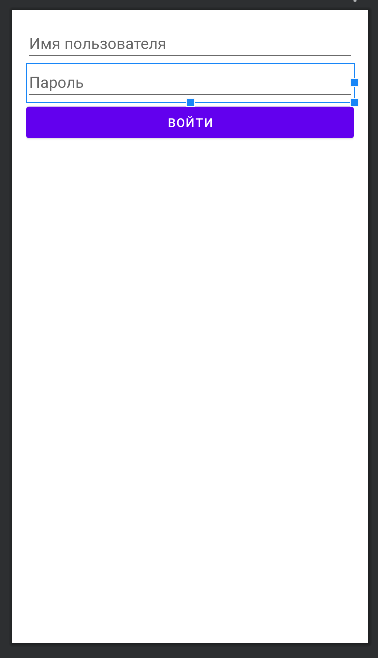


Рисунок 27 – Макет входа, функция «авторизация пользователя»

На рисунке 28 изображен макет формы «Поиск заказа» (activity\_main).

Окно содержит следующие основные компоненты: поле ввода (inputSearch)и кнопку "Найти" (button).

Для того чтобы вывести окно с информацие о заказе, необходимо ввести номер заказа в текстовое поле и нажать кнопку "Найти".

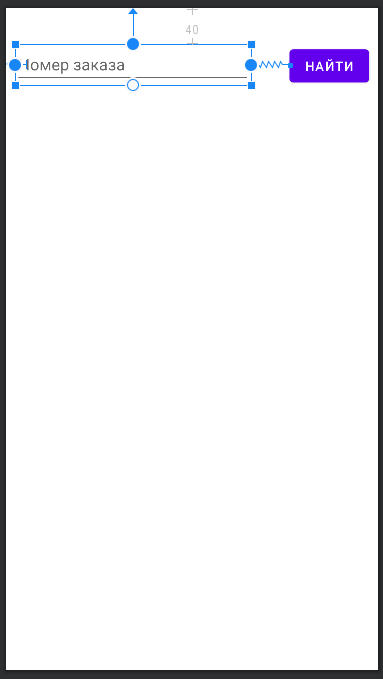


Рисунок 28 – Макет окна «Поиск заказа»

На рисунке 29 изображен макет формы «Информация о заказе» (activity\_edit\_order).

Окно содержит следующие основные компоненты: пять текстов и кнопку "статус заказа" (button3).

Для того чтобы изменить статус заказа, необходимо нажать на кнопку "Статус заказа".

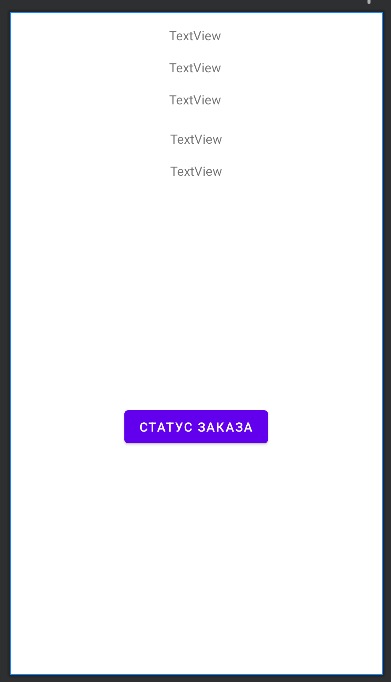


Рисунок 28 – Макет окна «Информация о заказе»

На рисунке 29 изображён макет отслеживания статуса заказа(activity\_main). Это окно видит пользователь, когда запускает приложение для клиентов.

Окно содержит следующие основные компоненты: поле ввода (inputSearch)и кнопку "Отследить" (button).

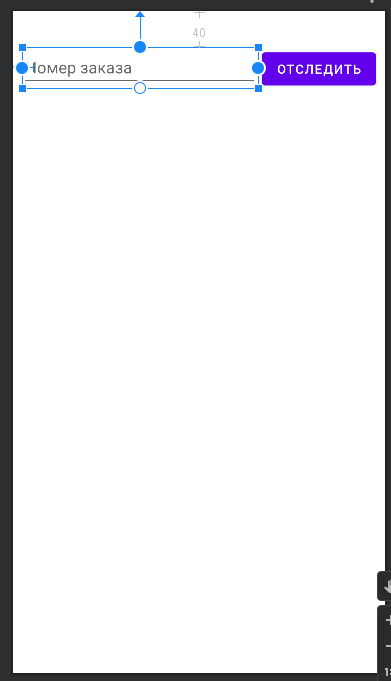


Рисунок 29 – Макет отслеживание статуса заказа, функция «поиск заказа»

На рисунке 30 изображен макет формы «Статус заказа» (activity\_track\_order).

Окно содержит следующие основные компоненты: шесть текстов и пять фигур.

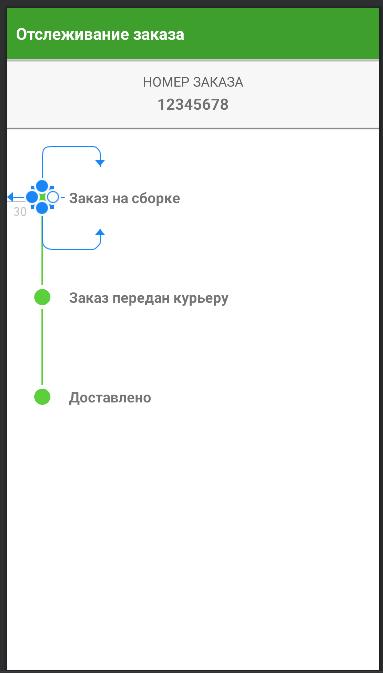


Рисунок 28 – Макет окна «Поиск заказа»

**3.2 Программная реализация**

В результате проведенной исследовательской работы, был разработан графический интерфейс, который эффективно взаимодействует с пользователем.

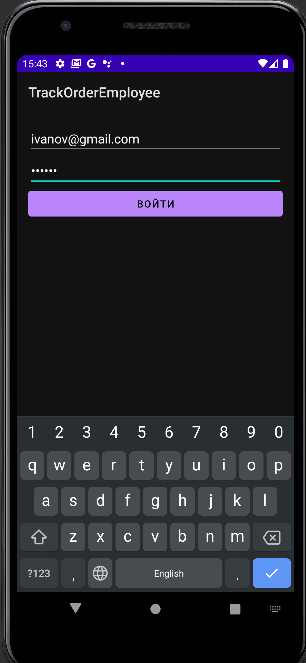


Рисунок 29 – Главное окно программы «Авторизация»

Для того, чтобы войти в приложение сотрудника магазина пользователю необходимо авторизироваться.

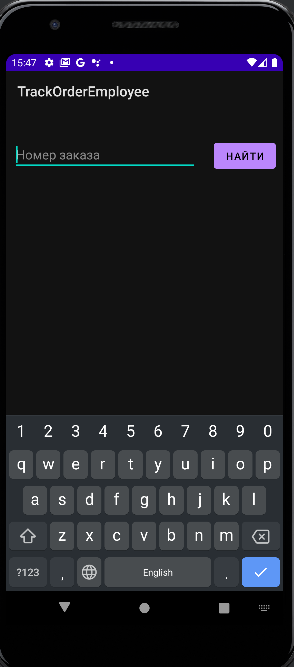


Рисунок 30 – Окно поиска заказа

Для перехода в окно с информацией о заказе необходимо ввести номер заказа.

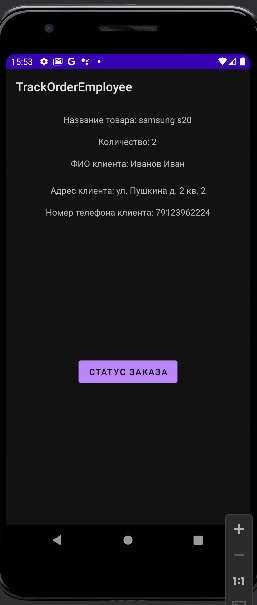


Рисунок 31 – Окно с информацие о заказе

После ввода номера заказа открывается окно с информацией о заказе.

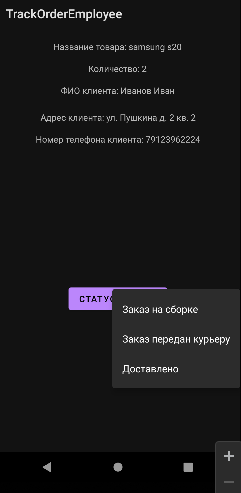


Рисунок 31 – Список статусов заказа

Для того, чтобы изменить статус заказа нужно нажать на кнопку «Статус заказа».

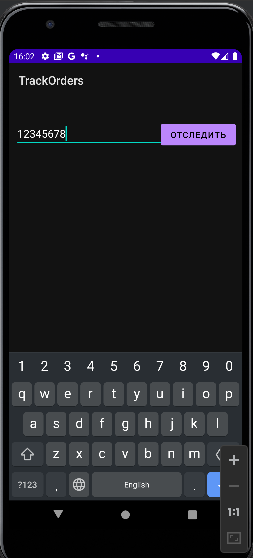


Рисунок 31 – Список статусов заказа

Для того, чтобы отследить статус заказа нужно ввести номер заказа и нажать на кнопку «Отследить».

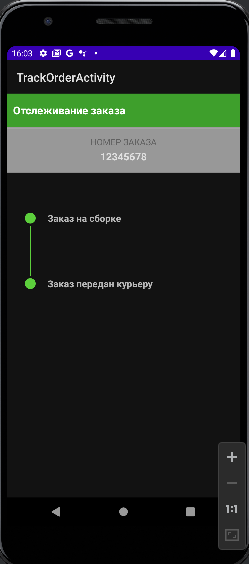


Рисунок 31 – Список статусов заказа

После ввода номера заказа открывается окно со статусом заказа.

Весь исходный код выложен в репозитории онлайн-сервиса Github https://github.com/wirgy/trackorder.

Среда программирования в данной работе, была использована  Android studio, на языке kotlin.

**Заключение**

В рамках нашей курсовой работы мы разработали приложение на языке kotlin, которое позволяет отслеживать статус заказа.

В процессе работы над проектом мы получили ценные навыки в формализации системных требований к программному продукту и в архитектурном и интерфейсном проектировании. Мы освоили передовые методологии, позволяющие построить модели бизнес-процессов и основных компонентов программного продукта, что обеспечило нам прочную основу для разработки приложения.

На текущем этапе разработки мы успешно формализовали системные требования и спроектировали всю систему. Мы создали привлекательный графический интерфейс приложения, который позволяет легко отслеживать статус заказа. Наше приложение предлагает интуитивно понятный пользовательский опыт.

**Список использованных источников**

1. Пример использования диаграмм UML для описание предметной области (документ Microsoft word). Составители: Бахвалова З.А.
2. Анализ ПО\_25.11.16 (презентация). Составители: Бахвалова З.А.
3. Курсовая работа «Разработка прикладного программного обеспечения» Руководство и методические указания для студентов направления: 09.03.02 «Информационные системы и технологии». Составитель Бахвалова З.А. – Иркутск, Изд-во ИРНИТУ, 2018, 61 с.
4. "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
5. "The Pragmatic Programmer: Your Journey to Mastery" by Andrew Hunt and David Thomas
6. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
7. "Refactoring: Improving the Design of Existing Code" by Martin Fowler
8. "Code Complete: A Practical Handbook of Software Construction" by Steve McConnell
9. "Domain-Driven Design: Tackling Complexity in the Heart of Software" by Eric Evans
10. "The Mythical Man-Month: Essays on Software Engineering" by Frederick P. Brooks Jr.
11. "Pragmatic Unit Testing in Java with JUnit" by Andy Hunt and Dave Thomas
12. "Clean Architecture: A Craftsman's Guide to Software Structure and Design" by Robert C. Martin
13. "Software Engineering: A Practitioner's Approach" by Roger S. Pressman