МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский экономический университет им. Г.В. Плеханова»

Московский приборостроительный техникум

**Курсовой проект**

ПМ 01 Разработка модулей программного обеспечения для компьютерных систем

МДК 01.01 Разработка программных модулей

Специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

Тема: «Разработка мобильного приложения для машазина кондитерской«Upavlika»

**Пояснительная записка**

Листов: 47

|  |  |
| --- | --- |
|  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / М.А.Горбунова  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 год  Исполнитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Билашенко П.А  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 год |

2025

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

«УТВЕРЖДАЮ»

Заместитель директора по учебной работе

Д.А. Клопов

« » 2025 г.

**ЗАДАНИЕ**

на выполнение курсового проекта (курсовой работы)

Билашенко Павлу Антоновичу

(фамилия, имя, отчество студента — полностью)

студенту группы П50-1-22 специальности 09.02.07 «Информационные системы и программирование» по МДК 01.01 «Разработка программных модулей»

1. Исходные данные к проекту (работе):
   1. Тема: «Разработка мобильного приложения «Ежедневник».
   2. Состав курсового проекта:
      1. Задание КП
      2. Пояснительная записка
      3. Программа (исходные данные) на электронном носителе
      4. Презентация и инсталляционный пакет программы на электронном носителе
   3. Содержание пояснительной записки:

ВВЕДЕНИЕ

1. ОБЩАЯ ЧАСТЬ
   1. Цель разработки
   2. Средства разработки
2. СПЕЦИАЛЬНАЯ ЧАСТЬ
   1. Постановка задачи
      1. Входные данные предметной области
      2. Выходные данные предметной области
      3. Требования к проекту
   2. Внешняя спецификация
      1. Описание задачи
      2. Входные и выходные данные
      3. Методы
      4. Тесты
      5. Контроль целостности данных
   3. Проектирование
      1. Схема архитектуры приложения
      2. Логическая схема данных
      3. Физическая схема данных
      4. Структурная схема
      5. Функциональная схема
      6. Диаграмма классов
      7. Схема тестирования
      8. Схема пользовательского интерфейса
   4. Результат работы программы
3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ
   1. Инструментальные средства
   2. Отладка программы
   3. Защитное программирование
   4. Характеристики и программы

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

ПРИЛОЖЕНИЕ А. Описание задачи

ПРИЛОЖЕНИЕ Б. Диаграмма классов

ПРИЛОЖЕНИЕ В. Структурная схема

ПРИЛОЖЕНИЕ Г. Сценарий и результаты тестовых испытаний

ПРИЛОЖЕНИЕ Д. Текст программы

ПРИЛОЖЕНИЕ Е. Техническое задание

ПРИЛОЖЕНИЕ Ж Руководство пользователя

1. Содержание задания по проекту (работе) — перечень вопросов, подлежащих разработке

|  |  |  |  |
| --- | --- | --- | --- |
|  | Разрабатываемый вопрос | Объем от всего задания, % | Срок выполнения |
| А | Описательная часть проекта (введение, общее описание и т. д.) | 5 | 17.02.2025 |
| 1. | Введение | - | 17.02.2025 |
| 2. | Цель разработки | - | 17.02.2025 |
| 3. | Средства разработки | - | 17.02.2025 |
| Б | Анализ задачи и её постановка | 15 | 24.02.2025 |
| 1. | Определение требований к программе | - | 18.02.2025 |
| 2. | Спецификация программы (описание задачи, описание входных и выходных данных, методы) | - | 18.02.2025 |
| 3. | Тесты, контроль целостности данных | - | 24.02.2025 |
| В | Проектирование и реализация | 55 | 15.03.2025 |
| 1. | Схемы проекта (схема архитектуры, логическая схема данных, физическая схема данных, функциональная и структурная схемы, диаграмма классов, схема тестирования, схема пользовательского интерфейса) | - | 18.02.2025 |
| 2. | Реализация в инструментальной среде | - | 15.03.2025 |
| Г | Технологическая часть проекта | 5 | 20.03.2025 |
| 1. | Инструментальные средства разработки | - | 20.03.2025 |
| 2. | Отладка программа | - | 20.03.2025 |
| 3. | Защитное программирование | - | 20.03.2025 |
| 4. | Характеристика программы | - | 20.03.2025 |
| Д | Программная документация | 10 | 01.04.2025 |
| 1. | Приложение А. Описание задачи | - | 01.04.2025 |
| 2. | Приложение Б. Диаграмма классов | - | 01.04.2025 |
| 3. | Приложение В. Структурная схема | - | 01.04.2025 |
| 4. | Приложение Г. Сценарий и результаты тестовых испытаний |  | 01.04.2025 |
| 5. | Приложение Д. Текст программы | - | 01.04.2025 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Разрабатываемый вопрос | Объем от всего задания, % | Срок выполнения |
| 6. | Приложение Ж. Руководство пользователя |  | 01.04.2025 |
|  | Приложение Е. Техническое задание |  | 01.04.2025 |
| Е | Экспериментальная часть проекта | 10 | 21.04.2025 |
| 1. | Программа на машинном носителе.  Информация на носителе разбита на разделы: эксплуатационный пакет, тексты программы, документация. | - | 21.04.2025 |

Руководитель курсового проекта (работы) Горбунова Мария Александровна, преподаватель

«13» 2025 года января /М.А.Горбунова /

Дата выдачи курсового задания «13» января 2025 года

Срок сдачи законченного проекта (работы) «20» апреля 2025 года Задание принял к исполнению

«13» января 2025 года / П.А.Билашенко /

Содержание

[1.1. Цель разработки 9](#_Toc196296306)

[1.2. Средства разработки 9](#_Toc196296307)

[2.1. Постановка задачи 11](#_Toc196296308)

[2.2. Входные данные 11](#_Toc196296309)

[2.3. Выходные данные 11](#_Toc196296310)

[2.4. Подробные требования к проекту 12](#_Toc196296311)

[2.5. Внешняя спецификация 13](#_Toc196296338)

[2.6. Проектирование 20](#_Toc196296344)

[2.7. Результат работы приложения 22](#_Toc196296350)

[3.1. Инструментальные средства разработки 28](#_Toc196296352)

[3.2. Защитное программирование 29](#_Toc196296353)

[3.3. Характеристики программы 32](#_Toc196296356)

[ПРИЛОЖЕНИЕ А. ОПИСАНИЕ ЗАДАЧИ 38](#_Toc196296357)

[ПРИЛОЖЕНИЕ Б. ДИАГРАММА КЛАССОВ 46](#_Toc196296358)

[ПРИЛОЖЕНИЕ В. ТЕКСТ ПРОГРАММЫ 50](#_Toc196296359)

[ПРИЛОЖЕНИЕ Г. СТРУКТУРНАЯ СХЕМА 210](#_Toc196296360)

[ПРИЛОЖЕНИЕ Д. СЦЕНАРИЙ И РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ 216](#_Toc196296361)

[ПРИЛОЖЕНИЕ Е. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 231](#_Toc196296369)

[ПРИЛОЖЕНИЕ Ж. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 264](#_Toc196296370)

ВВЕДЕНИЕ

В современных условиях цифровизации все более актуальной становится потребность в эффективных инструментах для онлайн-шоппинга, сравнения товаров и управления покупками. Развитие технологий открывает широкие возможности для создания решений, способствующих удобству приобретения товаров и повышению качества обслуживания клиентов.

Однако, несмотря на обилие интернет-магазинов и торговых платформ, пользователи сталкиваются с фрагментированностью функционала: одни сервисы предлагают широкий ассортимент, другие – удобную доставку, третьи – привлекательные цены, что вынуждает покупателей обращаться к нескольким платформам одновременно. Это снижает удобство использования и затрудняет процесс выбора и приобретения товаров.

В связи с этим особую значимость приобретает разработка универсального решения, объединяющего ключевые функции для комфортного онлайн-шоппинга в едином приложении.

Приложение «Upavlika» призвано устранить эту проблему, предлагая пользователям единое пространство для поиска товаров, сравнения цен, оформления заказов, отслеживания доставки и управления покупками. Такой подход не только оптимизирует процесс приобретения товаров, но и способствует формированию лояльности клиентов, что особенно важно для современных потребителей, ценящих удобство и экономию времени.

Таким образом, создание интегрированного интернет-магазина для комплексного удовлетворения потребностей покупателей представляется своевременным и востребованным решением, способным упростить процесс онлайн-шоппинга и повысить эффективность взаимодействия между продавцами и покупателями.

1. ОБЩАЯ ЧАСТЬ

## Цель разработки

Создание универсального приложения «ЕжедневникПлюс», объединяющего функции ведения дневника, трекинга настроения, управления задачами и социальными контактами, для повышения личной продуктивности и эмоционального благополучия пользователей.

## Средства разработки

Для разработки приложения, реализующего свой функционал на базе Android были использованы средства разработки, представленные в Таблице 1.

Таблица 1 – Технические средства

|  |  |  |
| --- | --- | --- |
| № | Тип оборудования | Наименование оборудования |
| 1 | 2 | 4 |
| Xiaomi 12 S | | |
| 1 | Процессор: | Helio G96 |
| 2 | Емкость аккумулятора: | 5000 мА\*ч |
| 3 | Интерфейсы: | USB Type-C |
| 4 | Оперативная память: | 6 ГБ |

В Таблице 2 представлены минимальные и рекомендованные технические средства, на базе которых возможно комфортное использование реализуемого приложения.

Таблица 2 – Конфигурация технических средств

|  |  |  |
| --- | --- | --- |
| № | Тип оборудования | Наименование оборудования |
| 1 | 2 | 4 |
| Минимальные технические требования | | |
| 1 | Процессор: | Exynos 4412 1,4 ГГц |
| 2 | Оперативная память: | 2 Гб |
| 3 | Внутреннее хранилище: | Не менее 100 Мб |
| 4 | Дисплей: | 720х1280 (HD) |
| 5 | Емкость аккумулятора: | 2100 мАч |
| № | Тип оборудования | Наименование оборудования |
| 6 | Интерфейсы: | Micro USB, Wi-Fi 802.11, GPS |
| 7 | Операционная система | Android не менее 8.0 |
| Рекомендуемые технологические требования | | |
| 1 | Процессор: | Qualcoмм Snapdragon 712 2,3 ГГц |
| 2 | Оперативная память: | 4 Гб |
| 3 | Внутреннее хранилище: | от 100 Мб |
| 4 | Дисплей: | 1080х1920 (FullHD) |
| 5 | Емкость аккумулятора: | 4000 мАч |
| 6 | Интерфейсы: | USB Type-C, Wi-Fi 802.11, GPS |
| 7 | Операционная система | Android не менее 8.0 |

Для разработки приложения использовались программные средства, представленные в Таблице 3.

Таблица 3 – Программные средства разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип | Наименование | Назначение |
| 1 | 2 | 3 | 4 |
| 1 | Инструментальное средство разработки мобильного приложения | AndroidStudio 2024.3.1.14 | Разработка мобильного приложения |
| 2 | Текстовый редактор | Microsoft Word 2019 | Разработка документации, формирование отчетных документов по шаблону. |
| 3 | Браузер | Google Chrome | Доступ к панели управления облачной базы данных. |

1. СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

Разработать мобильное приложение для онлайн-торговли с целью просмотра и покупки товаров, отслеживания заказов и доставки, управления учетной записью пользователя, включающее в себя облачную базу данных и программный интерфейс, реализующий взаимодействие базы данных с приложением.

## Входные данные

Входные данные представлены в следующем виде:

* Аутентификационные данные пользователя (Адрес электронной почты, Пароль);
* Персональные данные (Имя, Адрес, Контактный телефон, город, почтовый индекс);
* Данные запроса товаров (Категории);
* Данные корзины (Товар, Количество);
* Данные заказа (Товары);
* Данные администрирования (Информация о товаре и пользователе, Статус заказа).

## Выходные данные

Выходные данные представлены в следующем виде:

* Каталог товаров (Название, Изображение, Цена, Категория, Описание);
* Корзина покупок (Товар, Количество, Сумма, Общая стоимость);
* Данные заказа (Номер, Статус, Дата, Сумма, Товары);
* Данные отслеживания доставки (Статус,Ожидаемая дата доставки);
* История заказов (Список заказов, Статусы, Даты);
* Данные профиля пользователя (Имя, Контактная информация, История заказов);
* Данные администрирования (Список товаров, Список заказов, Список пользователей);
* Данные экспорта (Информация о заказах).

## Подробные требования к проекту

Необходимо разработать мобильное приложение, реализовав в составе нее следующие процессы:

* Проектирование мобильного приложения - необходимо спроектировать мобильное приложение интернет-магазина, программный интерфейс для обеспечения взаимодействия приложения и базы данных. Также, в процессе проектирования необходимо реализовать диаграмму прецедентов, схему бизнес-процессов IDEF0, архитектурную схему приложения, структурную схему, функциональную схему, диаграмму классов, схему тестирования;
* Подключение базы данных – необходимо реализовать хранение данных о товарах, заказах и пользователях, обеспечивающее валидацию и манипуляцию входными и выходными данными приложения. База данных должна предусматривать валидацию всех вводимых данных, шифрование паролей учетных записей пользователей;
* Разработка мобильного приложения на базе Android – реализующее функции интернет-магазина. Приложение должно предусматривать реализацию всех базовых функций, в том числе:

## Управление каталогом товаров:

## Просмотр списка товаров с возможностью фильтрации и сортировки

## Детальный просмотр информации о товаре

## Поиск товаров по названию и категориям

## Управление корзиной:

## Добавление товаров в корзину

## Изменение количества товаров

## Удаление товаров из корзины

## Расчет общей стоимости заказа

## Управление заказами:

## Оформление заказа с указанием адреса доставки

## Просмотр истории заказов

## Отслеживание статуса доставки заказа

## Получение уведомлений о статусе заказа

## Управление пользователями:

## Регистрация и авторизация пользователей

## Редактирование профиля пользователя

## Восстановление пароля через имя

## Просмотр истории заказов пользователя

## Административные функции:

## Управление каталогом товаров (добавление, редактирование, удаление)

## Управление заказами (просмотр, изменение статуса)

## Доступ к панели администратора для авторизованных пользователей с соответствующей ролью

## Дополнительные функции:

## Отслеживание статуса доставки в реальном времени

## Система уведомлений о статусе заказа и доставки

## Внешняя спецификация

### Описание задачи

Описание задачи приведено в приложении А «Описание задачи».

### Входные и выходные данные

В Таблице 4 представлены входные данные, вводимые пользователем в приложение.

Таблица 4 – Входные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Ограничение | Формат ввода | Описание |
| 1 | 2 | 3 | 4 | 5 |
| Форма аутентификации | | | | |
| Имя | Тип | Ограничение | Формат ввода | Описание |
| 1 | 2 | 3 | 4 | 5 |
| E-mail | string | name@domen.domen | Текстовое поле | Адрес электронной почты регистрирующегося пользователя |
| Password | string | [a-zA-Z0-9]{6, 25} | Текстовое поле | Пароль регистрирующегося пользователя |
| Персональные данные | | | | |
| Имя | Тип | Ограничение | Формат ввода | Описание |
| 1 | 2 | 3 | 4 | 5 |
| E-mail | string | name@domen.domen | Текстовое поле | Адрес электронной почты пользователя |
| Name | string | [а-яА-Яa-zA-Z0-9]{3, 20} | Текстовое поле | Имя пользователя |
| Телефон | string | +7[0-9]{10} | Текстовое поле | Контактный телефон пользователя |
| Адрес | string | [а-яА-Яa-zA-Z0-9,.]{5, 100} | Текстовое поле | Адрес доставки |
| Данные корзины | | | | |
| Количество | number | [1-10] | Текстовое поле/счетчик | Количество товара в корзине |
| Данные товаров для админа | | | | |
| Name | string | [а-яА-Яa-zA-Z0-9]{3, 50} | Текстовое поле | Название товара |
| Description | string | [а-яА-Яa-zA-Z0-9,.]{0, 500} | Многострочное текстовое поле | Описание товара |
| Price | decimal | > 0 | Числовое поле | Цена товара |
| Category | string | [а-яА-Яa-zA-Z0-9]{2, 30} | Текстовое поле / список | Категория товара |
| Данные отслеживания заказа | | | | |
| Номер заказа | string | [a-zA-Z0-9-]{5, 20} | Текстовое поле | Идентификатор заказа для отслеживания |

Входные данные представлены в следующем виде:

* Аутентификационные данные пользователя:
  + Адрес электронной почты
  + Пароль
* Регистрационные данные:
  + Имя
  + Адрес электронной почты
  + Пароль
  + Подтверждение пароля
* Данные для оформления заказа:
  + Адрес доставки
  + Статус оплаты
  + Статус заказа
* Данные товаров в корзине:
  + Количество
* Данные для управления товарами (админ):
  + Название
  + Описание
  + Цена
  + Категория

### Методы

При разработке мобильного приложения для управления личными данными и социальными взаимодействиями были применены следующие методы и подходы:

* **Архитектура:**
  + Традиционная Android-архитектура (активности + фрагменты)
  + Singleton для корзины и UserManager
  + DTO-модели для бизнес-логики
  + Разделение слоёв данных и представления
* **Работа с данными:**
  + SQLite через DatabaseHelper
  + ContentValues/Cursor для запросов
  + Транзакции для заказов
  + SharedPreferences для сессий
* **UI-компоненты:**
  + ConstraintLayout для вёрстки
  + RecyclerView с адаптерами
  + TabLayout для категорий
  + Кастомные диалоги
* **Безопасность:**
  + Аутентификация по email/паролю
  + Ролевая модель (админ/пользователь)
  + Валидация вводимых данных
* **Оптимизация:**
  + Фоновые Thread-ы для БД
  + Ленивая загрузка компонентов
  + Кэширование корзины
* **Модели данных:**
  + User (роли, баланс)
  + Product (категории, цены)
  + Order (статусы, оплата)
  + Cart (синглтон)
* **Состояние приложения:**
  + Восстановление при повороте
  + Intent для передачи данных
  + Связка с жизненным циклом
  + **Электронная коммерция:**
  + Фильтрация товаров
  + Корзина с редактированием
  + История заказов
  + Управление профилем
* **Обработка ошибок:**
  + Try-catch для БД/сети
  + Логирование через Log
  + Toast-уведомления
* **UX-решения:**
  + Интуитивное меню
  + Мгновенная валидация форм
  + Восстановление пароля
  + Сообщения об успешных операциях

### Тесты

Сценарий тестирования и результаты тестовых испытания приведен в приложении Г «Сценарий и результаты тестовых испытаний».

### Контроль целостности данных

В Таблице 5 представлен контроль целостности данных, описывающий ситуации и реакцию приложения на выполнение задач, связанных с сохранением, выводом, изменением или удалением данных.

Таблица 5 – Контроль целостности данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ситуация | Аномалия | Реакция | Примечание |
| 1 | Авторизация | Ввод несуществующей пары email-пароль | Сообщение «Неверный email или пароль» | Авторизация с неверными данными невозможна |
| 2 | Регистрация | Email уже зарегистрирован | Сообщение «Пользователь с таким email уже существует» | Уникальность email обязательна |
| 3 | Регистрация | Пароли не совпадают | Сообщение «Пароли не совпадают» | Поля «Пароль» и «Подтвердите пароль» должны совпадать |
| 4 | Регистрация | Пароль короче 6 символов | Сообщение «Пароль должен содержать не менее 6 символов» | Минимальная длина пароля — 6 символов |
| 5 | Добавление в корзину | Недостаток товара на складе | Добавление доступного количества + уведомление | Невозможно добавить больше, чем есть в наличии |
| 6 | Оформление заказа | Недостаточно средств на балансе | Сообщение «Недостаточно средств на балансе» | Оплата возможна только при достаточном балансе |
| 7 | Восстановление пароля | Пользователь не найден | Сообщение «Пользователь с таким именем не найден» | Восстановление доступно только для существующих аккаунтов |
| 8 | Изменение данных товара (админ) | Пустые название/цена | Сообщение «Заполните обязательные поля» | Товар требует названия и цены |
| 9 | Отмена заказа | Попытка отменить доставленный заказ | Сообщение «Нельзя отменить доставленный заказ» | Отмена доступна только для статусов: «Ожидает оплаты», «Оплачен», «Обработка» |
| 10 | Редактирование профиля | Некорректный формат телефона | Сообщение «Введите корректный номер телефона» | Требуется соответствие формату (например, +7 XXX XXX-XX-XX) |
| 11 | Редактирование корзины | Количество товара < 1 | Автоматическое удаление товара | Товар удаляется при нулевом/отрицательном количестве |
| 12 | Добавление пользователя (админ) | Email уже используется | Сообщение «Пользователь с таким email уже существует» | Уникальность email для каждого пользователя |
| 13 | Оплата заказа | Повторная оплата | Сообщение «Заказ уже оплачен» | Запрет двойной оплаты |
| 14 | Изменение статуса заказа | Недопустимое изменение (например, «Отменен» → «Обработка») | Сообщение «Недопустимое изменение статуса» | Строгая последовательность статусов |
| 15 | Пополнение баланса | Отрицательная сумма | Сообщение «Сумма должна быть положительной» | Запрет отрицательных значений |

## Проектирование

### Схема архитектуры приложения

На Рисунке 1 представлена архитектурная схема мобильного приложения.

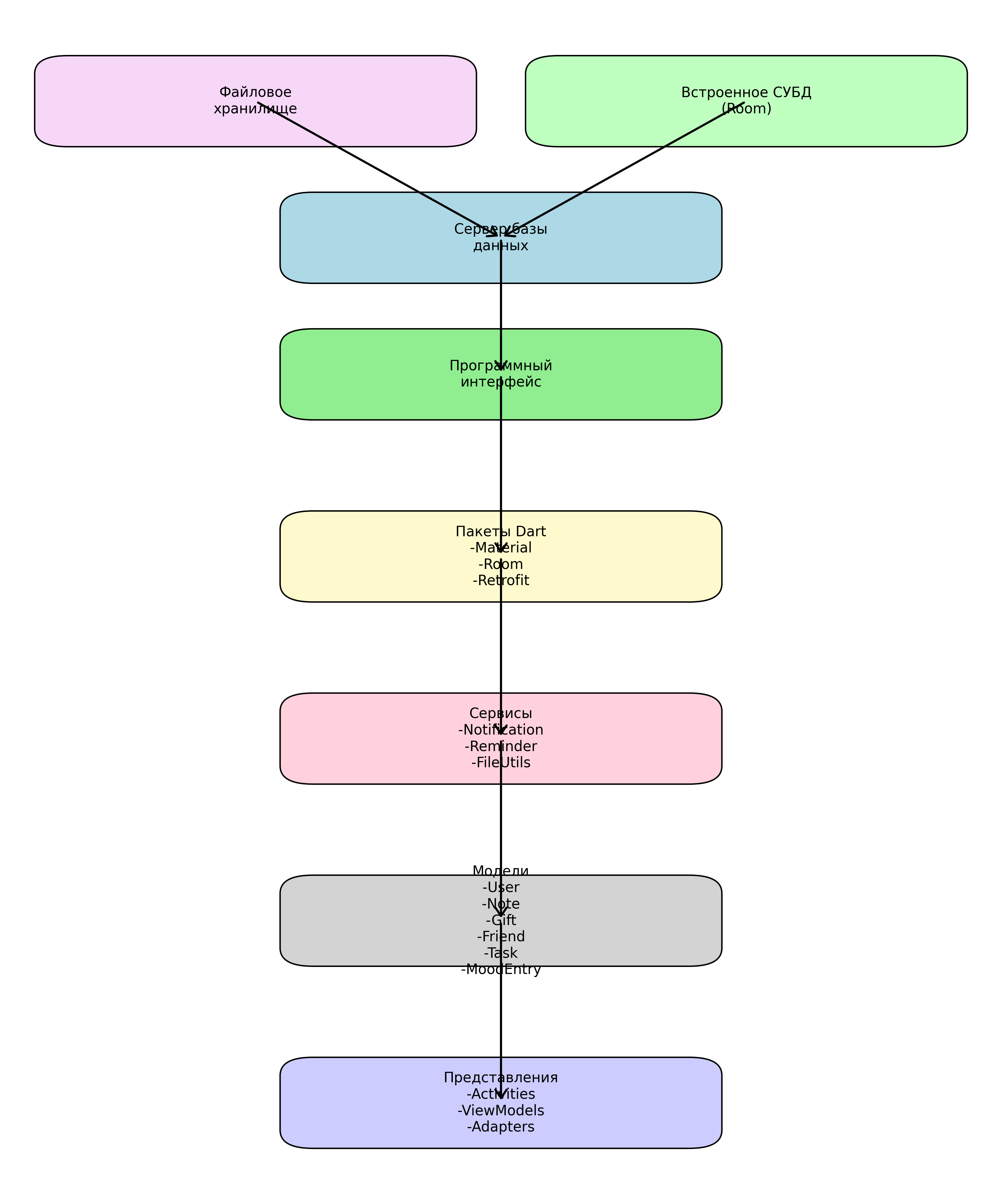


Рисунок 1 – Архитектурная схема приложения

Схема показывает, что архитектура реализуемой информационной системы представляет из себя распределенную Клиент-серверную архитектурную систему.

### Структурная схема

Структурная схема представлена в приложении В «Структурная схема».

### Диаграмма классов

Диаграмма классов представлена в приложении Б «Диаграмма классов».

### Схема тестирования

Схема и методики тестирования представлены в приложении Г «Сценарий и результаты тестовых испытаний».

### Схема пользовательского интерфейса

На Рисунке 2 продемонстрирована схема интерфейса мобильного приложения. На данной схеме продемонстрированы все экраны и процесс навигации между ними.

Изображение выглядит как зарисовка, диаграмма, Технический чертеж, План

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 – Схема пользовательского интерфейса приложения

## Результат работы приложения

В В результате выполнения курсовой работы было разработано мобильное приложение для Android, предназначенное для онлайн-заказа хлебобулочных и кондитерских изделий. Приложение реализовано на языке Java с использованием современных инструментов разработки и архитектурных подходов.В качестве локальной базы данных была выбрана SQLite с использованием встроенного в Android фреймворка, что позволило обеспечить надежное хранение данных на устройстве пользователя и удобную работу с ними. База данных содержит несколько основных сущностей: пользователи, товары, заказы, позиции заказов и профили пользователей. Структура базы данных оптимизирована для быстрого доступа к информации и эффективного использования памяти устройства.В приложении реализован полноценный функционал электронной коммерции, включающий просмотр каталога товаров по категориям, детальную информацию о каждом товаре, добавление товаров в корзину и оформление заказов. Пользователи могут отслеживать статус своих заказов и просматривать историю покупок.Особое внимание было уделено пользовательскому интерфейсу, который разработан в соответствии с принципами Material Design и обеспечивает интуитивно понятное взаимодействие с приложением. Реализована удобная система поиска товаров по названию и описанию, что позволяет быстро находить нужные позиции в каталоге.В приложении внедрена система управления пользователями с различными ролями (администратор и обычный пользователь), что позволяет разграничить права доступа к функциональности. Администраторы имеют возможность управлять каталогом товаров, обрабатывать заказы и изменять их статусы.Для обеспечения безопасности реализована система аутентификации пользователей с возможностью регистрации новых аккаунтов и авторизации существующих. Пользователи могут управлять своим профилем, включая контактную информацию и адрес доставки.Архитектура приложения построена с использованием паттерна MVC с четким разделением логики представления, бизнес-логики и доступа к данным, что обеспечивает поддержание и масштабирование кода. Для работы с базой данных использован паттерн Singleton, обеспечивающий единую точку доступа к хранилищу данных.В приложении реализована система учета баланса пользователей и механизм оплаты заказов, позволяющий клиентам отслеживать свои расходы и статус оплаты. Предусмотрена возможность отмены заказов с указанием причины.В процессе разработки особое внимание было уделено обработке ошибок, что позволило создать стабильное приложение, способное корректно работать даже в нестандартных ситуациях.Тестирование приложения проводилось на различных версиях Android и разных устройствах, что позволило обеспечить стабильную работу на широком спектре мобильных телефонов. В результате было создано полнофункциональное мобильное приложение для заказа хлебобулочных изделий, отвечающее современным требованиям к пользовательскому опыту и производительности. Приложение успешно решает поставленные задачи по организации онлайн-заказов и предоставляет пользователям удобный и интуитивно понятный инструмент для покупки кондитерских изделий.

На рисунках 3-5 продемонстрирована работа приложения.

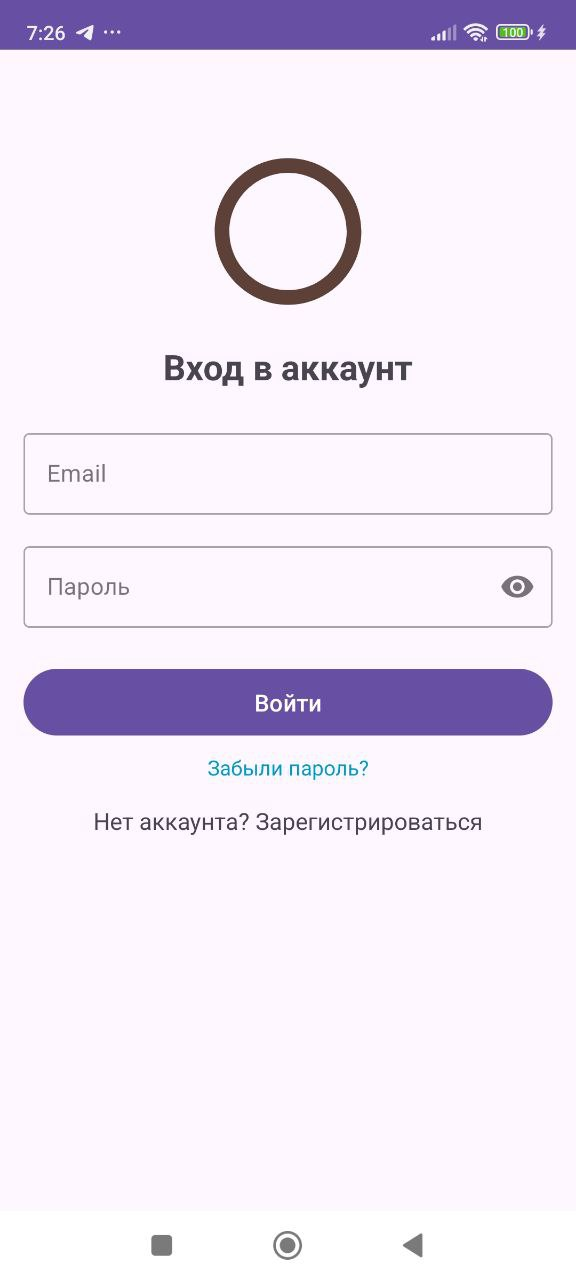


Рисунок 3 – Окно авторизации

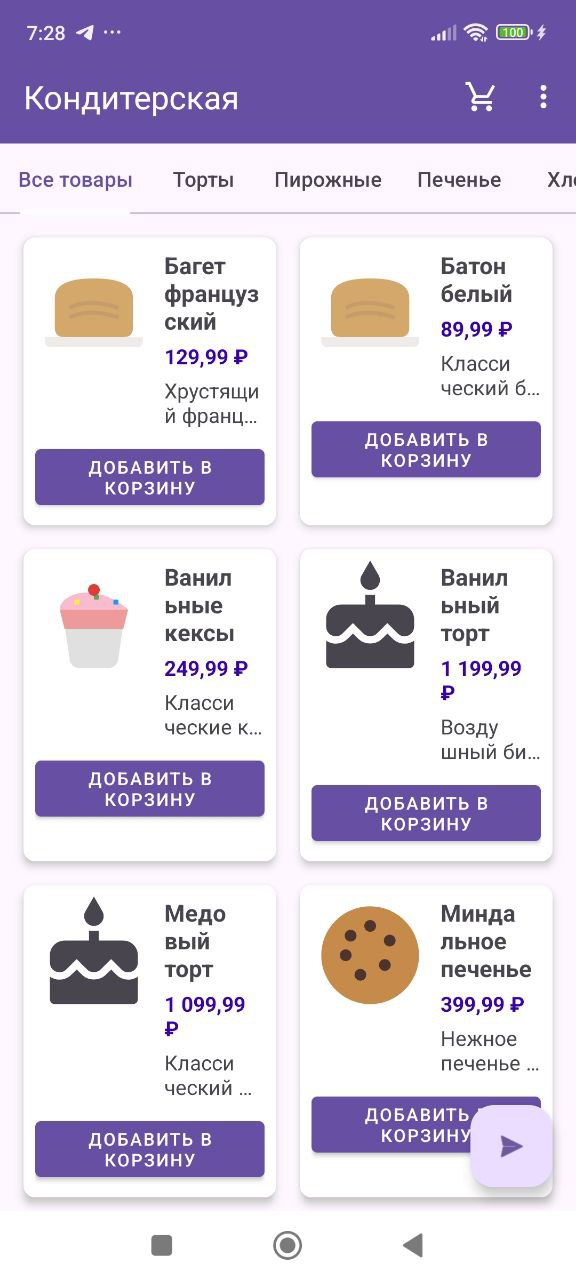


Рисунок 4 – Главный экран

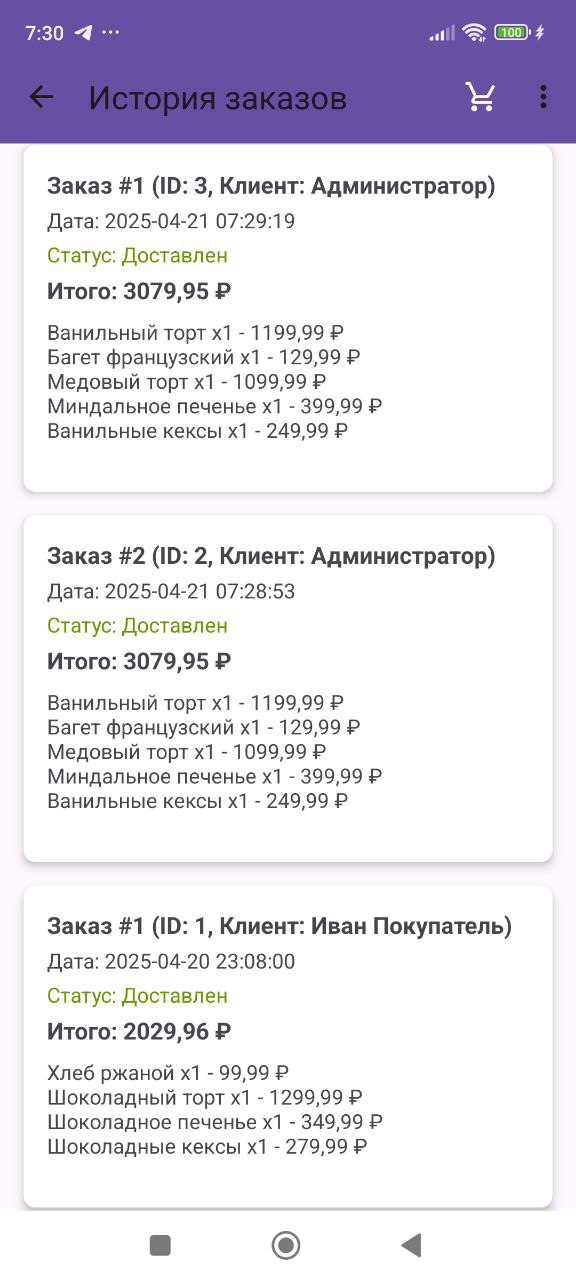


Рисунок 5 – Страница «История заказов»

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ



## Инструментальные средства разработки

При разработке мобильного приложения для онлайн-заказа хлебобулочных и кондитерских изделий использованы проверенные технологии. Основной язык программирования — Java, выбранный благодаря стабильности, богатой экосистеме и обширной документации для Android-разработки.

Для хранения данных применена встроенная СУБД SQLite — легковесное решение, не требующее отдельного сервера. Управление базой данных реализовано через SQLiteOpenHelper, обеспечивающий создание, обновление структуры и подключение к БД.

База данных включает следующие таблицы:

users — данные пользователей (учётные записи, баланс);

products — каталог товаров (название, описание, цена, количество);

orders — информация о заказах (статус, дата, сумма);

order\_items — позиции заказов (количество, цена товаров);

user\_profiles — контактные данные для доставки.

Интегрированная среда разработки — Android Studio, предоставляющая полный инструментарий:

Встроенный эмулятор устройств;

Инструменты профилирования производительности;

Интеграция с системой контроля версий (Git);

LogCat для мониторинга логов.

Пользовательский интерфейс построен на стандартных компонентах Android UI и элементах Material Design, что обеспечивает единообразие дизайна и соответствие рекомендациям Google.

Архитектура приложения основана на паттерне MVC:

Model: Классы для работы с БД (DatabaseHelper) и бизнес-сущности (User, Product, Order);

View: XML-макеты и компоненты Activity/Fragment;

Controller: Логика обработки действий пользователя в Activity и Fragment.

Для асинхронных операций (работа с БД, сетевые запросы) использованы AsyncTask и стандартные потоки Java, что сохраняет отзывчивость интерфейса.

Отладка приложения

Для отладки и тестирования приложения использовались как встроенные инструменты Android Studio, так и специализированные утилиты, такие как Android Debug Bridge (ADB) для взаимодействия с устройствами и эмуляторами. Тестирование проводилось на различных версиях Android с использованием как эмуляторов, так и реальных устройств.

## Защитное программирование

### Защита от ошибок

В процессе разработки мобильного приложения особое внимание было уделено обработке различных типов ошибок и исключительных ситуаций для обеспечения стабильной работы.В приложении реализована система обработки ошибок при работе с SQLite базой данных. Операции с базой данных выполняются с использованием try-catch блоков для перехвата и корректной обработки исключений SQLiteException, что позволяет избежать краха приложения при проблемах с доступом к данным.

java

Apply to DatabaseHelp...

try {

    // Код для работы с базой данных

    db.beginTransaction();

    // Операции вставки/обновления

    db.setTransactionSuccessful();

} catch (SQLiteException e) {

    Log.e("DatabaseHelper", "Ошибка в транзакции: " + e.getMessage(), e);

    // Обработка ошибки

} finally {

    db.endTransaction();

}

Особое внимание уделено обработке null-значений. В коде используются проверки на null перед операциями с объектами, что позволяет избежать NullPointerException:

java

Apply to DatabaseHelp...

if (cursor != null && cursor.moveToFirst()) {

    // Работа с курсором

    cursor.close();

}

При работе с пользовательскими данными применяется валидация входных параметров, что помогает предотвратить ввод некорректных данных. Реализована проверка наличия пользователя перед выполнением операций обновления или удаления:

java

Apply to DatabaseHelp...

User user = getUserByEmail(email);

if (user == null) {

    return false;  // Пользователь не существует

}

В приложении реализована система логирования с использованием стандартного класса Log, что позволяет отслеживать и анализировать возникающие проблемы. Логи содержат информацию о выполняемых операциях и возникающих исключениях:

java

Apply to DatabaseHelp...

Log.d("DatabaseHelper", "Заказ успешно создан с ID: " + orderId);

Log.e("DatabaseHelper", "Ошибка при вставке товара заказа: " + item.getProductName());

Для операций с заказами и платежами реализован механизм транзакций, обеспечивающий атомарность операций и предотвращающий неконсистентность данных при возникновении ошибок:

java

Apply to DatabaseHelp...

db.beginTransaction();

try {

    // Операции с базой данных

    db.setTransactionSuccessful();

} finally {

    db.endTransaction();

}

Система обновления базы данных использует версионирование, что обеспечивает безопасное обновление структуры базы данных при выпуске новых версий приложения:

java

Apply to DatabaseHelp...

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    if (oldVersion < 2) {

        // Добавление новых таблиц или столбцов

    }

}

Таким образом, реализованная система защиты от ошибок обеспечивает стабильную работу приложения в различных условиях использования и предоставляет разработчику информацию для отладки и улучшения приложения.

### Защита данных

Защите также подвержены данные в базе данных, в виде невозможности создать 2 аккаунта с одним адресом электронной почты. Так как это уникальное значение.

## Характеристики программы

Разработанное мобильное приложение "upavlika" представляет собой нативное Android-приложение, предназначенное для работы на устройствах под управлением операционной системы Android версии 7.0 (API level 24) и выше. Такой выбор минимальной версии Android обеспечивает поддержку более 90% активных Android-устройств на рынке, при этом позволяя использовать современные API и функциональные возможности платформы.Приложение оптимизировано для работы на различных типах устройств с разными размерами экранов и разрешениями дисплея. Благодаря использованию адаптивного дизайна и компонентов Material Design (com.google.android.material), интерфейс приложения корректно масштабируется на смартфонах, обеспечивая комфортное использование независимо от размера экрана устройства.Архитектура приложения построена с учетом особенностей платформы Android и оптимизирована для эффективного использования системных ресурсов. Использование SQLite в качестве локальной базы данных (через DatabaseHelper) обеспечивает быстрый доступ к данным и эффективное управление памятью устройства. Приложение использует современные компоненты пользовательского интерфейса, такие как SwipeRefreshLayout для обновления контента и ConstraintLayout для создания отзывчивых интерфейсов.Приложение поддерживает работу как в онлайн, так и в офлайн режиме. При отсутствии подключения к интернету пользователь может продолжать работу с локально сохраненными данными в базе SQLite. Это обеспечивает непрерывность работы пользователя независимо от качества сетевого подключения.Система уведомлений реализована с использованием сервисов Android, включая DeliveryTrackerService и DeliveryNotificationReceiver, что позволяет получать уведомления о статусе заказов даже когда приложение не активно. При этом учитываются особенности работы с уведомлениями на различных версиях Android, обеспечивая корректное отображение и обработку уведомлений на всех поддерживаемых устройствах.Приложение эффективно использует системные ресурсы устройства, минимизируя потребление памяти и энергии. Реализованы механизмы освобождения ресурсов при переходе приложения в фоновый режим, а также оптимизация работы с изображениями и ресурсами для снижения нагрузки на процессор и батарею устройства.Приложение включает полный функционал для управления заказами, профилями пользователей, корзиной покупок и отслеживания доставки. Для администраторов предусмотрена специальная панель управления (AdminPanelActivity), позволяющая управлять товарами и пользователями.Приложение включает в себя систему авторизации и регистрации пользователей, возможность просмотра и редактирования профиля, каталог товаров с детальной информацией о каждом продукте, корзину для оформления заказов, историю заказов, а также систему отслеживания доставки в реальном времени с использованием геолокации.Благодаря использованию современных компонентов Android и следованию рекомендациям по разработке для платформы Android, приложение обеспечивает плавную анимацию и отзывчивый интерфейс даже на устройствах среднего ценового сегмента. Время запуска приложения оптимизировано за счет эффективного управления жизненным циклом активностей и фрагментов.Таким образом, разработанное приложение полностью соответствует современным требованиям к Android-приложениям, обеспечивая высокую производительность, надежность и удобство использования для клиентов сервиса доставки хлебобулочных и кондитерских изделий.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта было разработано мобильное приложение для платформы Android, предназначенное для электронной коммерции и управления заказами. В ходе работы были успешно решены все поставленные задачи и достигнуты намеченные цели проекта.В процессе разработки были изучены и применены современные технологии и инструменты разработки мобильных приложений, включая язык программирования Java, среду разработки Android Studio и систему управления базами данных SQLite. Особое внимание было уделено реализации паттерна Singleton для работы с базой данных, что позволило эффективно управлять состоянием приложения и обеспечить целостность данных.Разработанное приложение предоставляет пользователям широкий спектр функциональных возможностей, включая просмотр каталога товаров, управление корзиной покупок, оформление заказов, отслеживание статуса доставки и управление личным профилем. Реализованная система аутентификации обеспечивает безопасный доступ к функциям приложения с разграничением прав между обычными пользователями и администраторами.В ходе работы над проектом были успешно решены технические задачи по организации хранения данных, обеспечению безопасности пользовательской информации и оптимизации производительности приложения. Использование современных программных подходов позволило создать хорошо структурированный и легко поддерживаемый код.Особое внимание было уделено пользовательскому интерфейсу приложения, который разработан в соответствии с принципами Material Design и обеспечивает интуитивно понятное взаимодействие с различными функциями приложения. Использование компонентов TextInputLayout, RecyclerView и CardView позволило создать современный и отзывчивый интерфейс, адаптированный для различных размеров экранов.В процессе разработки были успешно преодолены различные технические сложности, связанные с управлением жизненным циклом активностей, обработкой пользовательского ввода и обеспечением целостности данных в локальной базе. Реализованная структура базы данных с уникальными ограничениями для email пользователей обеспечивает защиту от дублирования учетных записей.Проведенное тестирование приложения подтвердило его стабильность и надежность. Особое внимание было уделено проверке функциональности оформления заказов, учета товаров и взаимодействия с пользовательским профилем, что позволило создать качественный программный продукт, готовый к практическому использованию.Разработанное приложение имеет потенциал для дальнейшего развития и масштабирования. В будущем возможно добавление новых функций, таких как интеграция с платежными системами, расширенная аналитика продаж, внедрение механизмов рекомендаций и программы лояльности для пользователей.Таким образом, в рамках курсового проекта было создано полнофункциональное мобильное приложение для электронной коммерции, которое успешно решает поставленные задачи по управлению товарами и заказами. Приложение соответствует современным требованиям к мобильным приложениям и готово к дальнейшему развитию и совершенствованию.

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

1. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.

2. ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.

3. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.

4. ГОСТ Р 7.0.5-2008 БИБЛИОГРАФИЧЕСКАЯ ССЫЛКА. Общие требования и правила составления.

5. Android Developers – Официальная документация [Электронный ресурс]. – URL: https://developer.android.com/docs

6. Android Studio User Guide – Руководство по IDE [Электронный ресурс]. – URL: https://developer.android.com/studio/intro

7. Material Design Guidelines – Руководство по дизайну [Электронный ресурс]. – URL: https://m3.material.io/

8. Android Jetpack – Библиотеки для разработки [Электронный ресурс]. – URL: https://developer.android.com/jetpack

9. GitHub – Система контроля версий [Электронный ресурс]. – URL: https://github.com/

10. Stack Overflow – Сообщество разработчиков [Электронный ресурс]. – URL: https://stackoverflow.com/

11. Android Developers Blog – Блог разработчиков Android [Электронный ресурс]. – URL: https://android-developers.googleblog.com/