

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
Федерального государственного бюджетного образовательного учреждения высшего
образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

КУРСОВАЯ РАБОТА

По Разработка приложений для мобильных операционных систем

Тема «Магазин б/у телефонов»

Руководитель

Колпаков А.А

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Шеварев Д.А.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

В данной курсовой работе рассматривается разработка и анализ информационной системы для магазина продажи поддержанных телефонов. В условиях растущего интереса к вторичному рынку мобильных устройств, эффективное управление информацией становится ключевым фактором для оптимизации бизнес-процессов и улучшения обслуживания клиентов. Целью работы является создание мобильного приложения на языке Kotlin с использованием среды разработки Android Studio, которое обеспечит учет товаров, отслеживание продаж, управление клиентскими данными и генерацию отчетов. В рамках работы решаются задачи по проектированию архитектуры системы, разработке базы данных, созданию пользовательского интерфейса и реализации функциональных возможностей приложения.

This thesis focuses on the development and analysis of an information system for a store specializing in the sale of used mobile phones. Given the growing interest in the secondary market for mobile devices, effective information management is crucial for optimizing business processes and improving customer service. The goal of the work is to create a mobile application using Kotlin and the Android Studio development environment, which will provide inventory management, sales tracking, customer data management, and report generation. The tasks include designing the system architecture, developing the database, creating the user interface, and implementing the functional capabilities of the application

Содержание

Введение.....	6
1. Анализ технического задания.....	8
1.1 Описание предметной области.....	8
1.2 Формирование требований к системе.....	9
1.3 Обоснование выбора средств разработки.....	9
2. Разработка алгоритмов.....	12
2.1 Концептуальная модель базы данных.....	12
2.2 Логическая модель базы данных.....	13
2.3 Физическая модель базы данных.....	14
2.4 Разработка и реализация приложения	15
2.5 Тестирование АИС	18
3. Руководство программиста.....	20
4. Руководство пользователя.....	24
Заключение.....	26
Список используемой литературы.....	27
Приложение 1.....	28
Приложение 2.....	30
Приложение 3.....	31

					МИВлГУ 09.03.04 - 0.029						
Изм.	Лист	№ докум.	Подпись	Дата							
Разраб.		Шеварев Д.А.			«Магазин б/у телефонов»			Лит.	Лист	Листов	
Провер.		Колпаков А.А.								5	36
Реценз.								МИ ВлГУ ПИН-121			
Н. Контр.											
Утверд.											

Введение

В условиях современного потребительского рынка, где мобильные телефоны становятся неотъемлемой частью повседневной жизни, растет интерес к вторичному рынку этих устройств. Потребители ищут качественные поддержанные телефоны по более доступным ценам, что делает магазины, занимающиеся продажей таких устройств, особенно актуальными. Эффективное управление информацией в таких магазинах становится необходимостью, так как это позволяет не только оптимизировать бизнес-процессы, но и улучшить уровень обслуживания клиентов.

Информационные системы (ИС) играют ключевую роль в управлении торговыми точками, специализирующимися на продаже поддержанных телефонов. Они обеспечивают учет товара, отслеживание продаж, управление клиентскими данными и генерацию отчетов. В результате использование ИС позволяет более эффективно реагировать на потребности клиентов и адаптироваться к изменениям рынка.

Кроме того, в современном мире, где не у каждого есть доступ к персональному компьютеру, но практически у каждого имеется мобильное устройство, становится актуальным создание приложений, которые могут работать на смартфонах и планшетах. Это позволяет значительно расширить доступность информационной системы для сотрудников магазина и клиентов, обеспечивая удобство и мобильность в управлении бизнес-процессами. Мобильные приложения могут предоставлять возможность оперативного доступа к данным о товарах, продажах и клиентах, что особенно важно в условиях динамично меняющегося рынка.

Целью данной курсовой работы является разработка и анализ информационной системы для магазина продажи поддержанных телефонов. В рамках работы планируется решить следующие задачи:

- Исследование бизнес-процессов магазинов, специализирующихся на продаже поддержанных телефонов.

					МИВлГУ 09.03.04 – 0.029	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

- Проектирование базы данных, отражающей специфику данного рода предприятий.
- Разработка функционала системы для работы с пользователем.

					МИВлГУ 09.03.04 – 0.029	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1. Анализ технического задания.

В данной курсовой работе необходимо разработать информационную систему магазина продажи б/у телефонов, занимающегося продажей мобильных телефонов. Реализуемая программа должна быть разработана с помощью Android Studio. Android Studio является одной из наиболее популярных и мощных платформ для разработки мобильных приложений. Она предоставляет разработчикам широкий набор инструментов и возможностей для создания масштабируемых, высокопроизводительных и безопасных приложений.

1.1 Описание предметной области

Информационная система «Магазин продажи поддержанных телефонов» предназначена для автоматизации процессов, связанных с продажей мобильных телефонов, которые уже были в употреблении. В этой системе пользователи взаимодействуют с программой через различные функциональные модули, каждый из которых обеспечивает определенные функции и задачи.

Основные объекты предметной области:

Клиенты: - Клиенты могут зарегистрироваться в системе, создавать учетные записи, связывая их с покупками и продажами.

- Зарегистрированные пользователи могут просматривать доступные модели телефонов, а также могут продать свое устройство. Могут просматривать историю своих покупок.

Администраторы: - Администраторы могут зарегистрироваться в системе, просматривать историю продаж, редактировать и удалять товары, а также редактировать роли пользователя.

База данных (БД) будет является основой информационной системы и обеспечит хранение, обработку и доступ к информации.

					МИВлГУ 09.03.04 – 0.029	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

1.2 Формирование требований к системе

- Магазин занимается продажей телефонов бывших в употреблении, а также принятием устройств на реализацию от клиента
- База данных должна содержать следующие элементы: марка и модель телефона, основные характеристики, цена приема и продажи, фотографии.
- Должна быть возможность регистрации и авторизации пользователей в системе. Которые будут продавать и соответственно покупать имеющиеся в наличии устройства. Администраторы будут следить за продажами устройств, а также за модерацией существующих устройств.

1.3 Обоснование выбора средств разработки

Для разработки приложения курсового проекта была выбрана среда разработки Android Studio. Для разработки серверной части был выделен язык программирования: Kotlin + Java.

К достоинствам Android Studio относятся: - среда разработки поддерживает работу с несколькими языками программирования, к которым относятся самые популярные – C/C++, Java, Kotlin. - редактор кода, с которым удобно работать; - позволяет разрабатывать приложения не только для смартфонов/планшетов, а и для портативных ПК, приставок для телевизоров Android TV, устройств Android Wear, новомодных мобильных устройств с необычным соотношением сторон экрана;

- тестирование корректности работы новых игр, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе;
- рефакторинг уже готового кода;
- достаточно большая библиотека с готовыми шаблонами и компонентами для разработки ПО;

					МИВлГУ 09.03.04 – 0.029	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

- разработка приложения для Android N – самой последней версии операционной системы;
- предварительная проверка уже созданного приложения на предмет ошибок в нем;
- большой набор средств инструментов для тестирования каждого элемента приложения, игры;
- для неопытных/начинающих разработчиков специально создано руководство по использованию Android Studio, размещенное на официальном сайте утилиты.

Android Studio — это интегрированная среда разработки (IDE), предназначенная для создания приложений на платформе Android. Она предоставляет разработчикам мощные инструменты для написания, отладки и тестирования приложений, включая редакторы кода, визуальные редакторы интерфейса и встроенные средства для управления зависимостями.

Основные компоненты Android Studio:

Проектная структура: Android Studio использует гибкую структуру проектов, где файлы и ресурсы организованы по каталогам. Основные каталоги включают:

- src/main/java — содержит код приложения.
- src/main/res — включает ресурсы, такие как изображения, строки, макеты и стили.
- src/main/AndroidManifest.xml — файл, определяющий основные характеристики приложения, такие как его компоненты и разрешения.

Модули: Каждый проект может содержать несколько модулей, включая основное приложение и библиотечные модули. Это позволяет разделять функциональность и переиспользовать код. - Activity и Fragment:

- Activity — это компонент, представляющий один экран пользовательского интерфейса. Он отвечает за взаимодействие с пользователем и управляет жизненным циклом интерфейса.

- Fragment — это часть интерфейса, которая может быть встроена в Activity. Фрагменты позволяют создавать многоразовые компоненты и адаптировать интерфейс под различные размеры экранов.

View и Layout: - View — это базовый класс для всех компонентов пользовательского интерфейса, таких как кнопки, текстовые поля и изображения.

- Layout — это контейнеры, которые определяют, как элементы пользовательского интерфейса размещаются на экране. Существуют различные типы макетов, такие как LinearLayout, RelativeLayout и ConstraintLayout, каждый из которых имеет свои особенности. ViewModel и LiveData:

- ViewModel — это компонент архитектуры, который хранит и управляет данными, связанными с пользовательским интерфейсом, обеспечивая сохранение состояния при изменениях конфигурации.

- LiveData — это оборачиваемый класс данных, который позволяет компонентам интерфейса реагировать на изменения данных в реальном времени, обеспечивая более плавное взаимодействие с пользователем.

Библиотеки и зависимости: Android Studio поддерживает управление зависимостями с помощью системы Gradle. Это позволяет легко интегрировать сторонние библиотеки и модули, расширяющие функциональность приложения.

Android Studio предоставляет разработчикам все необходимые инструменты для создания качественных приложений, обеспечивая простоту разработки, отладки и тестирования. Понимание структуры и компонентов

					МИВлГУ 09.03.04 – 0.029	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

2. Разработка моделей данных

2.1 Концептуальная модель базы данных

Первый этап создания автоматизированной информационной системы - разработка концептуальной модели. При создании концептуальной модели базы данных мы демонстрируем область предмета, для которой предназначена система. Однако концептуальная модель не является идеальной для разработки, поэтому необходимо нормализовать все таблицы для создания реляционной базы данных. Следующим этапом является построение логической модели, которая аналогична концептуальной.

Модель учитывает все аспекты исследуемой предметной области, что позволяет получить точный результат. Далее необходимо создать физическую модель данных, которая определит, как данные будут храниться и доступны на физическом уровне: названия таблиц и столбцов, типы данных, первичные и внешние ключи и т.д.

Первым шагом в разработке концептуальной модели данных является выделение сущностей, а затем создание самой модели. Эта модель нужна для дальнейшего разрабатывания базы данных и её преобразования, например, в реляционную базу данных. Результат представлен на рисунке 1 в приложении 1.

2.2 Логическая модель базы данных

Графическое представление логической структуры исследуемой предметной области - цель построения логической модели. Благодаря этому моделируются сущности предметной области и их взаимоотношения. Сущности представляют объекты, на которые направлена деятельность в данной предметной области, а также субъекты, осуществляющие эту деятельность. Свойства объектов и субъектов описываются при помощи атрибутов. Взаимоотношения между объектами и субъектами показываются при помощи связей. Правила и ограничения взаимоотношений определяются свойствами-связями. Обычно связи отражают зависимости между сущностями и влияние одной сущности на другую.

На рисунке 2 в приложении 1 представлена логическая модель для информационной системы магазина б/у телефонов. Сущность «Телефоны» определяет всю информацию о телефонах, которые имеются в наличии или были проданы (удалены из БД). В сущности «Продажи» содержатся данные, связанные с продажей товаров, она связывает несколько сущностей: «Клиент» и «Телефоны».

					МИВлГУ 09.03.04 – 0.029	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

2.3 Физическая модель базы данных

Физическая модель данных, представленная на рисунке 3 в приложении 1, составлена на основе логической модели, где уже отражены и учтены все основные аспекты исследуемой предметной области. После проектирования этой модели будет осуществляться непосредственно создание базы данных и ее таблиц

Опираясь на данные разработанной диаграммы сущность-связь выделим шесть таблиц.

Первая таблица называется «Телефоны», в которой будет храниться информация о телефонах.

- Dom(ID) = {строка цифр}(Id)
- Dom(Производитель) = {строка символов}(Brand)
- Dom(Модель) = {строка символов}(Model)
- Dom(Характеристики) = {строка символов}(Description)
- Dom(Описание) = {строка символов}(Text)
- Dom(Статус) = {значение bool}(Status)
- Dom(Цена покупки) = {строка цифр}(Price)
- Dom(Цена продажи) = {строка цифр}(Magprice)
- Dom(Фото) = {строка символов}(Photo)

Вторая таблица называется «Продажи», в которой будет храниться информация о существующих транзакциях.

- Dom(ID продажи) = {строка цифр}(Id)
- Dom(Товар) = { FK }(Product)
- Dom(продавец) = { FK }(User)
- Dom(покупатель) = { FK }(SaleUser)
- Dom(цена покупки) = {строка цифр}(price)
- Dom(цена продажи) = {строка цифр}(magprice)
- Dom(Дата продажи) = {дата и время}(SaleDate)

Третья таблица называется «Пользователь», в которой будет храниться информация о существующих пользователях и их роли.

- Dom(ID пользователя) = { строка цифр }(Id)
- Dom(Email) = { строка символов }(Email)
- Dom(Логин) = { строка символов }(Login)
- Dom(Пароль) = { строка символов }(Password)
- Dom(Список товаров) = { FK }(Product)
- Dom(is_admin) = { значение bool }(isadmin)

2.4 Разработка и реализация приложения

Разработка и внедрение автоматизированной информационной системы (АИС) для магазина б/у телефонов является значимой задачей, требующей решения. Реализация проекта начинается с экрана авторизации и соответственно регистрации пользователя. Экран авторизации будет содержать поля для ввода email и пароля и кнопки “ Авторизация ” (Приложение 3 рисунок 4). В случае, если аккаунта у пользователя нет, он может зарегистрироваться, перейдя по ссылке ниже кнопки “ Авторизация ”(Приложение 3 рисунок 5).

Для разработки данных форм была использована таблица users:

```
CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    email TEXT,
    login TEXT,
    phone TEXT,
    pass TEXT,
    is_admin INTEGER
)
```

Данная таблица необходима для реализации ролей в информационной системе.

Для работы с базой данных был создан класс DBHelper, в котором реализован основной функционал: создание базы данных, отправка запросов, получение данных.

Были реализованы методы addUser, getUser, getUserById, getUserIdByEmail, isEmailUnique, getAllUsers, updateUserRole, addItem, getItems, getItemsById,

updateItemStatus, addSale, getSales, getSalesByUserId, updateItem, getItemById, getAllItems, deleteItem для работы с базой данных.

Формы авторизации и регистрации:

Форма авторизации (AuthActivity) содержит поля для ввода email и пароля, а также кнопку "Войти". При нажатии на кнопку "Авторизация" вызывается метод getUser из DBHelper, который проверяет наличие пользователя в базе данных. Если пользователь существует, он авторизуется и переходит на главную страницу (ItemsActivity). Если пользователь не найден, выводится сообщение об ошибке.

Форма регистрации (RegActivity) содержит поля для ввода логина, email, телефона, пароля и чекбокс для выбора роли администратора. При нажатии на кнопку "Зарегистрироваться" вызывается метод addUser из DBHelper, который добавляет нового пользователя в базу данных. Если email уже зарегистрирован, выводится сообщение об ошибке.

Главная страница и просмотр товаров:

После успешной авторизации пользователь попадает на главную страницу (ItemsActivity), где отображаются все доступные товары (Приложение 3 рисунок 6). Товары отображаются в RecyclerView с использованием адаптера ItemsAdapter. Пользователь может просматривать товары, а также использовать поиск для фильтрации товаров по бренду, модели или описанию.

При нажатии на товар открывается страница товара (ItemActivity) (Приложение 3 рисунок 7), где отображаются подробные сведения о товаре. Если пользователь не является администратором, он может нажать кнопку "Купить", чтобы приобрести товар. При нажатии на кнопку "Купить" вызывается метод addSale из DBHelper, который добавляет новую запись в таблицу sales и в историю покупок пользователя.

Профиль пользователя:

Пользователь может перейти на страницу профиля (ProfActivity), где отображаются его личные данные, список его товаров и история покупок. (Приложение 3 рисунок 8,9) Если пользователь является администратором, он может перейти на панель администратора (AdminActivity).

Панель администратора:

Администратор имеет доступ к панели управления (AdminActivity) (приложение 3 рисунок 10) где он может управлять пользователями, товарами и продажами. Администратор может просматривать всех пользователей (AdminUserActivity) (Приложение 3 рисунок 11), все товары (AdminProductActivity) (Приложение 3 рисунок 12) и все продажи (AdminSaleActivity) (Приложение 3 рисунок 13).

Администратор может редактировать существующие товары (EditItemActivity) и удалять товары. Для добавления нового товара используется метод addItem из DBHelper, для редактирования товара — метод updateItem, для удаления товара — метод deleteItem.

История покупок:

Пользователь может просматривать историю своих покупок на странице истории покупок (PurchaseHistoryActivity) (Приложение 3 страница 14). История покупок отображается в RecyclerView с использованием адаптера PurchaseHistoryAdapter. Для получения истории покупок используется метод getSalesByUserId из DBHelper.

2.4 Тестирование АИС

Основная цель проведения тестирования состоит в проверке соответствия реализации системы требуемой функциональности. Во время тестирования каждый тест регистрируется и его результаты сравниваются с ожидаемым. Если ожидаемый результат не совпадает с фактическим, это отмечается в протоколе тестирования. В таблице 1 представлена методика проведения тестирования разработанного программного продукта.

Выполненное действие	Полученный результат
Запуск исполняемого файла	Отображение формы регистрации.
Переход по ссылке	Отображение формы авторизации.
Заполнение данных	Вывод сообщения “пользователь успешно создан”
Заполнение существующими данными	Вывод сообщения "пользователь уже существует!"
Авторизация под именем «admin»	Отображение интерфейса администратора.
Изменение товара	Отображение новых данных в форме
Удаление товара	Удаление данных из таблицы.
Изменение роли пользователя	После авторизации пользователь может зайти на страницу администратора

Таблица 1 – методика тестирования разработанной программы

Отклонение транзакции	Изменение статуса транзакции на "Отклонено"
Авторизация пользователя	Отображение интерфейса пользователя. Возможность покупки товара
Нажатие кнопки «купить»	Удаление данного товара из списка товаров. В профиле пользователя добавляется история покупок, а также обновляется статус на «Товаро продан!». Отображение данных в т. Продаж у администратора
Нажатие на кнопку «продать устройство»	Переход на страницу заполнения данных
Нажатие на кнопку «продать»	Добавление товара в истории продаж пользователя. Добавление товара на главную страницу

Продолжение таблицы 1

Результаты, полученные в ходе тестирования разработанного программного продукта, позволяют сделать заключение в том, что разработанная программа соответствует требованиям технического задания.

3 Руководство программиста

Приложение включает в себя работу с базой данных, аутентификацию и регистрацию пользователей, а также функциональность для продажи клиентами телефонов и соответственно их покупки.

Основные компоненты приложения:

ItmesActivity: главная активность, с которой пользователь может переместиться в профиль, либо просмотреть список телефонов и найти интересующий его телефон.

ItemActivity: экран для отображения отдельно взятого телефона.

AuthActivity: экран авторизации пользователя.

RegActivity: экран регистрации пользователя.

ProfActivity: экран для отображения профиля пользователя.

ProfileItemsActivity: экран, отображающий список товаров, проданных пользователем.

ItemsAdapter: класс для работы с выводом данных об устройстве.

ProfileItemsAdapter: класс для работы с выводом данных о проданном устройстве в профиле пользователя.

User: модель пользователя.

Item: модель телефона.

DBHelper: класс для работы с базой данных SQLite.

Основные методы и функции:

DBHelper:

- 1) addUser(user: User): Добавляет нового пользователя в базу данных.
- 2) getUser(email: String, pass: String): Проверяет наличие пользователя с указанными email и паролем.
- 3) getUserById(userId: Int): Возвращает пользователя по его ID.
- 4) getUserIdByEmail(email: String): Возвращает ID пользователя по его email.
- 5) getUserIdByProductId(productId: Int): Возвращает ID пользователя по ID продукта.

- 6) `isEmailUnique(email: String)`: Проверяет уникальность email.
- 7) `getAllUsers()`: Возвращает список всех пользователей.
- 8) `updateUserRole(userId: Int, isAdmin: Boolean)`: Обновляет роль пользователя (администратор или нет).
- 9) `addItem(item: Item, userId: Int)`: Добавляет новый товар в базу данных.
- 10) `getItems()`: Возвращает список всех товаров.
- 11) `getItemsByUserId(userId: Int)`: Возвращает список товаров, проданных конкретным пользователем.
- 12) `updateItemStatus(itemId: Int, status: Boolean)`: Обновляет статус товара (продан или нет).
- 13) `addSale(sale: Sale)`: Добавляет новую продажу в базу данных.
- 14) `getSales()`: Возвращает список всех продаж.
- 15) `getSalesByUserId(userId: Int)`: Возвращает список продаж, совершенных конкретным пользователем.
- 16) `updateItem(item: Item)`: Обновляет информацию о товаре.
- 17) `getItemById(itemId: Int)`: Возвращает товар по его ID.
- 18) `getAllItems()`: Возвращает список всех товаров.
- 19) `deleteItem(itemId: Int)`: Удаляет товар из базы данных.

AuthActivity:

- 1) `onCreate(savedInstanceState: Bundle?)`: Инициализирует активность авторизации, устанавливает обработчики событий для кнопок и полей ввода.

RegActivity:

- 1) `onCreate(savedInstanceState: Bundle?)`: Инициализирует активность регистрации, устанавливает обработчики событий для кнопок и полей ввода.

ItemsActivity:

- 1) `onCreate(savedInstanceState: Bundle?)`: Инициализирует главную активность, устанавливает обработчики событий для кнопок и полей ввода, загружает список товаров.
- 2) `filterItems(query: String?)`: Фильтрует список товаров по заданному запросу.

ItemActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность отображения отдельно взятого товара, устанавливает обработчики событий для кнопок.

ProfActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность профиля пользователя, устанавливает обработчики событий для кнопок и полей ввода, загружает информацию о пользователе и его товарах.

ProfileItemsActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность отображения списка товаров, проданных пользователем.

ItemsAdapter:

1) onCreateViewHolder(parent: ViewGroup, viewType: Int): Создает новый ViewHolder для отображения элемента списка товаров.

2) onBindViewHolder(holder: MyViewHolder, position: Int): Привязывает данные товара к ViewHolder.

3) updateItems(newItems: List<Item>): Обновляет список товаров и уведомляет адаптер о изменениях.

ProfileItemsAdapter:

1) onCreateViewHolder(parent: ViewGroup, viewType: Int): Создает новый ViewHolder для отображения элемента списка проданных товаров.

1) onBindViewHolder(holder: MyViewHolder, position: Int): Привязывает данные проданного товара к ViewHolder.

AdminActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность администратора, устанавливает обработчики событий для кнопок.

AdminProductActivity

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность администратора для управления товарами, загружает список товаров.

2) onResume(): Обновляет список товаров при возобновлении активности.

AdminSaleActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность администратора для управления продажами, загружает список продаж.

AdminUserActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность администратора для управления пользователями, загружает список пользователей.

EditItemActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность редактирования товара, устанавливает обработчики событий для кнопок и полей ввода.

2) updateItem(itemId: Int): Обновляет информацию о товаре в базе данных.

SaleActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность добавления нового товара, устанавливает обработчики событий для кнопок и полей ввода.

2) addItem(): Добавляет новый товар в базу данных.

PurchaseHistoryActivity:

1) onCreate(savedInstanceState: Bundle?): Инициализирует активность отображения истории покупок пользователя, загружает список продаж.

Руководство пользователя для мобильного приложения

Это руководство поможет вам освоиться с основными функциями и страницами Android-приложения. В нем описаны ключевые действия, которые доступны пользователю.

Рекомендуемые системные требования:

Операционная система Android: 8.0.0;

Процессор с частотой 1 ГГц;

4 ГБ оперативной памяти;

Не менее 50 МБ места на жёстком диске; Наличие уже существующей базы данных не обязательно.

При запуске программы пользователь попадает на страницу регистрации (Приложение 3 рисунок 5), заполняет данные, далее нажимает на кнопку «Регистрация». После этого клиент получит сообщение об успешной регистрации, и перейдя по ссылке ниже попадает на страницу авторизации (Приложение 3 рисунок 4). Далее необходимо ввести данные, указанные при регистрации и нажать на кнопку «Авторизация». После нажатия на кнопку пользователь увидит сообщение об успешной авторизации и попадет на страницу с товарами (Приложение 3 рисунок 6). Пользователь может найти какой-либо товар, введя в поисковую строку название или модель интересующего устройства (Приложение 3 рисунок 15). Далее пользователь может выбрать товар, и после нажатия на кнопку «Просмотреть» пользователь попадет на страницу «Информация о товаре» (Приложение 3 рисунок 7), где указана основная информация о выбранном устройстве. Далее пользователь может нажать на кнопку «Купить», после чего вылетит подтверждение о покупке, которое можно подтвердить или отклонить, нажав на соответствующую кнопку. После подтверждения покупки товар будет удален из списка продаваемых устройств, а пользователь может перейти на страницу своего профиля нажав на кнопку в правом верхнем углу (Приложение 3

					МИВлГУ 09.03.04 – 0.029	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

рисунок 8). В профиле пользователь может увидеть список проданных им товаров, если они есть. А также две кнопки: «История покупок» и «Продать устройство». При нажатии на кнопку «История покупок» (Приложение 3 рисунок 14) пользователь попадет на страницу, где указан список купленных им товаров. Если нажать на кнопку «Продать устройство», то пользователь попадет на страницу заполнения данных об устройстве (Приложение 3 рисунок 16), и после заполнения может нажать на кнопку «Продать устройство», устройство появится в списке проданных.

Регистрация и авторизация администратора практически не отличается от пользователя, за исключением того, что при регистрации необходимо поставить галочку на «Регистрация как администратор». После успешной авторизации администратор может перейти в профиль, где присутствует основная информация о текущем пользователе и кнопка «Панель администратора» (Приложение 3 рисунок 9). При нажатии на данную кнопку пользователь перейдет на страницу с выбором разделов (Приложение 3 рисунок 10). Выбрав нужный раздел мы можем просмотреть какую либо информацию. При переходе в раздел «Пользователи» (Приложение 3 рисунок 11) мы сможем увидеть список зарегистрированных пользователей и поменять им роль, поставив галочку напротив «Роль администратора?». При переходе в раздел «Продажи» (Приложение 3 рисунок 13), мы сможем просмотреть подробную информацию о проданных магазином устройствах. При переходе в раздел «Телефоны» (Приложение 3 рисунок 12) мы увидим список всех телефонов. На странице каждого устройства есть две кнопки «Редактировать» и «Удалить». При нажатии на кнопку «Удалить» данное устройство удаляется из списка. При нажатии на кнопку «Редактировать» пользователь попадет на страницу редактирования. После изменения нужных полей и нажатии на кнопку «Редактировать товар» мы сможем увидеть в списке всех товаров изменения.

Заключение

В заключение хотелось бы отметить, что данная курсовая работа, разработанная на Android Studio, была предназначена для организации эффективного управления бизнес-процессами в данной сфере.

В ходе выполнения работы был осуществлен анализ требований к системе, проведено проектирование базы данных на SQLite, разработаны необходимые функциональные модули и интерфейсы на Android Studio, а также проведено тестирование и отладка системы.

Информационная система магазина б/у телефонов, основанная на сочетании SQLite и Android Studio, обладает рядом преимуществ. Прежде всего, данная связка технологий обеспечивает надежное хранение данных в базе данных, эффективное извлечение и обработку информации, а также обеспечивает удобный пользовательский интерфейс для взаимодействия с системой.

Разработанная информационная система позволяет упростить и автоматизировать процессы продажи б/у телефонов.

В результате выполнения данной работы была достигнута поставленная цель - разработка информационной системы, способной повысить эффективность работы магазина б/у телефонов. Однако, для ее дальнейшего успешного использования и развития, рекомендуется провести дополнительное тестирование системы на более широком объеме данных, а также реализовать функции взаимодействия с другими информационными системами.

В целом, данная работа демонстрирует возможности и перспективы использования Android Studio для разработки информационных систем, а также подчеркивает значимость автоматизации бизнес-процессов для повышения эффективности и конкурентоспособности организации.

Список литературы:

1. Введение в разработку приложений для смартфонов на ОС Android / А.Семакова Национальный открытый Университет «ИНТУИТ», 2016
2. Колисниченко Д.Н. Программирование для Android 5. Самоучитель. — СПб.: БХВ-Петербург, 2015. — 303 с.
3. 2016. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е изд. — СПб.: Питер, Гриффитс Дэвид, Гриффитс Дон Head First. Программирование для Android. 2-е изд. — СПб.: Питер, 2018. — 912 с.

					МИВлГУ 09.03.04 – 0.029	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

Приложение 1.

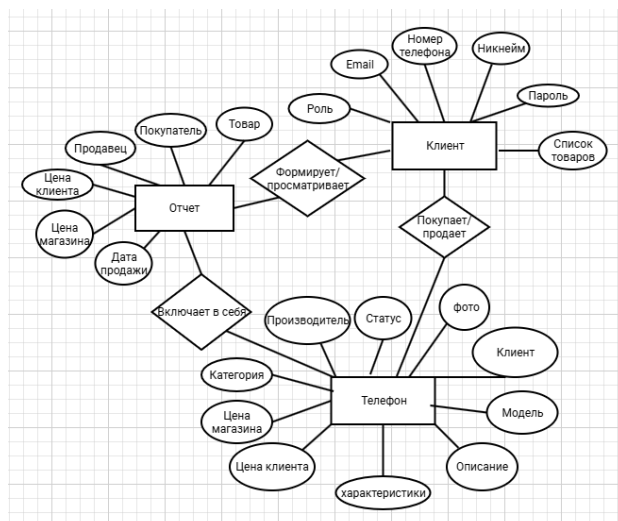


Рисунок 1 - Концептуальная модель данных

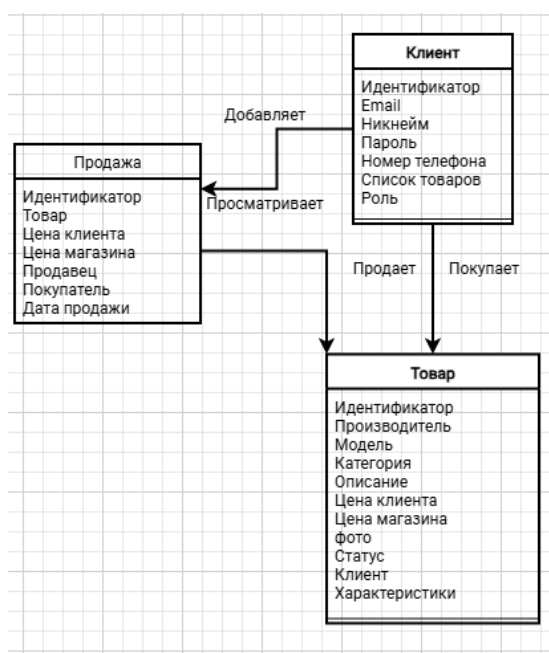


Рисунок 2 – Логическая модель данных

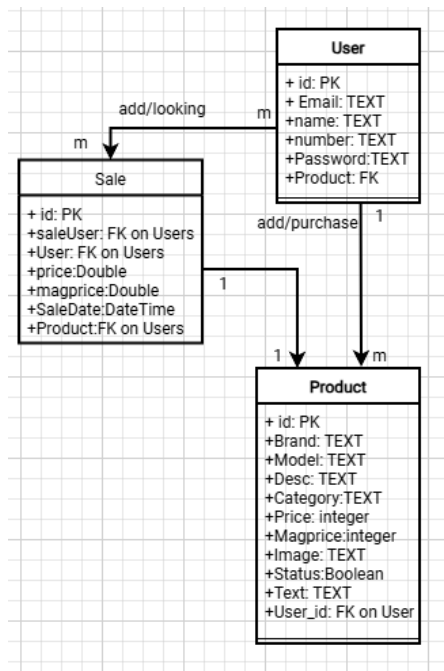


Рисунок 3 – Физическая модель данных

Приложение 2

С подробным кодом программы вы можете ознакомиться по данной ссылке:

<https://github.com/SHeFF33/kurosovayaAndroidStudio.git>

					МИВлГУ 09.03.04 – 0.029	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

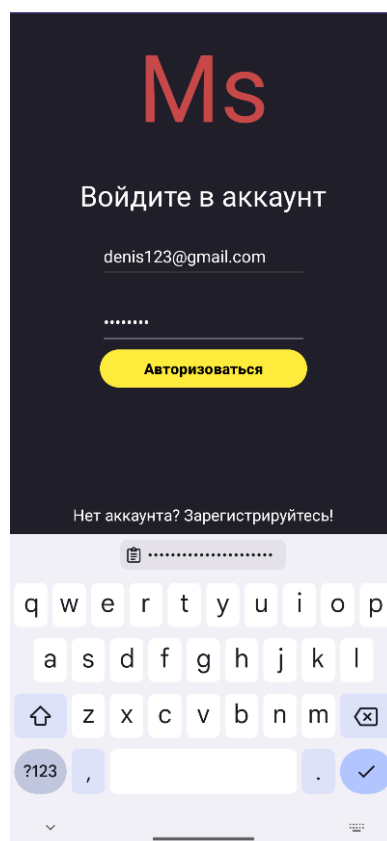


Рисунок 4 – Экран авторизации

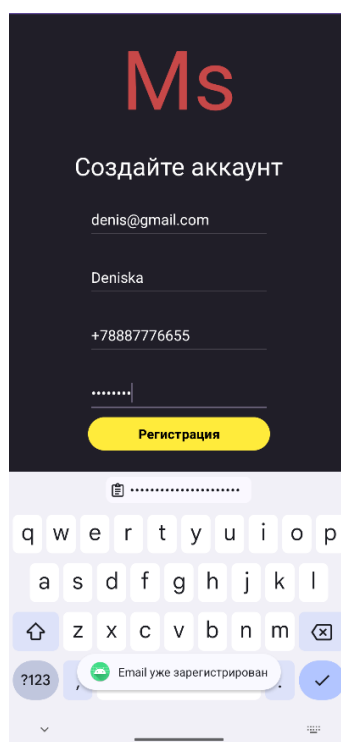


Рисунок 5 – Экран регистрации

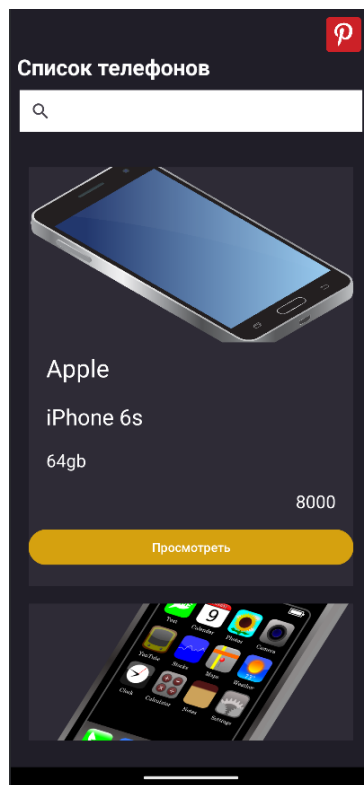


Рисунок 6 – Главный экран со списком товаров



Рисунок 7 – Экран просмотра товара

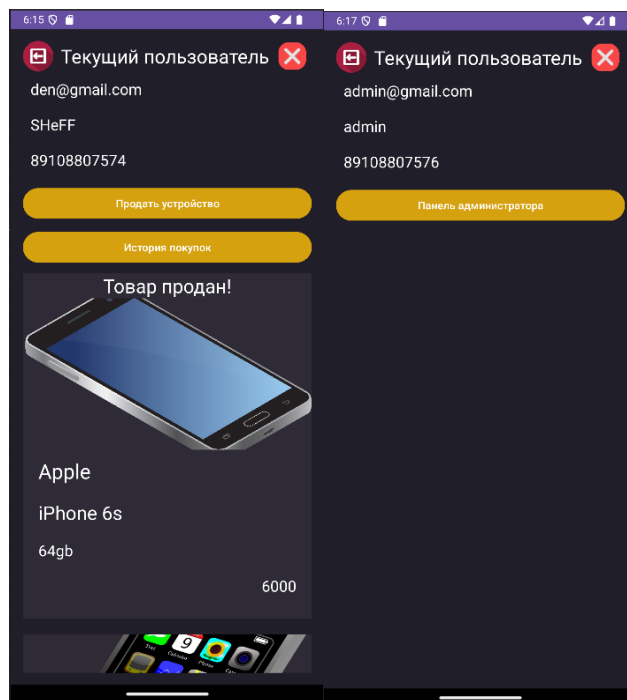


Рисунок 8, 9 – Профиль пользователя

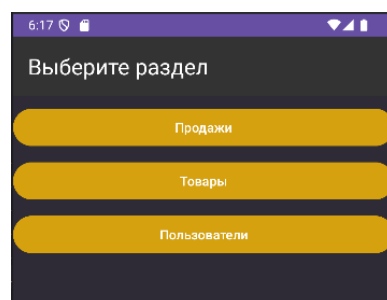


Рисунок 10 – Панель админа

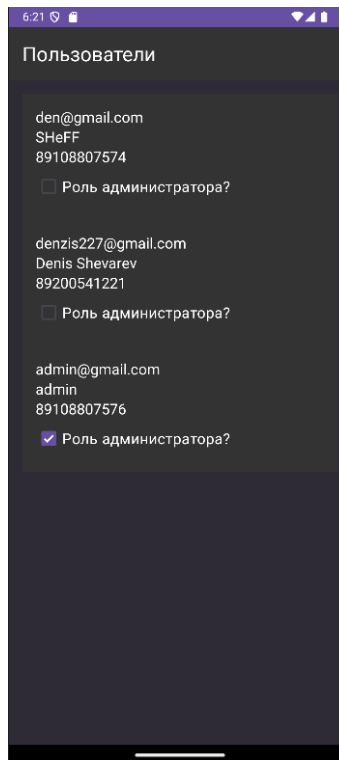


Рисунок 11 – Список всех пользователей

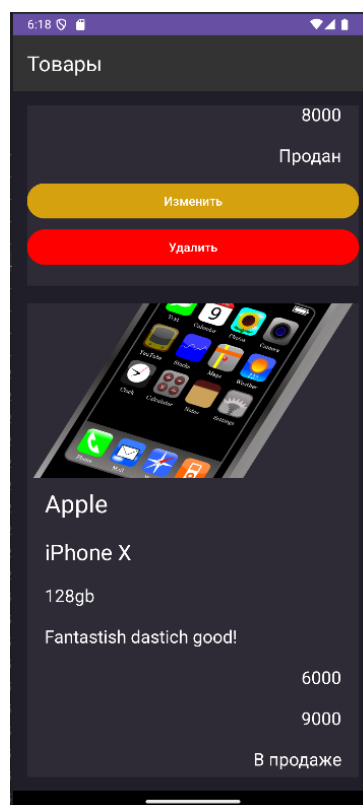


Рисунок 12 – Список всех товаров

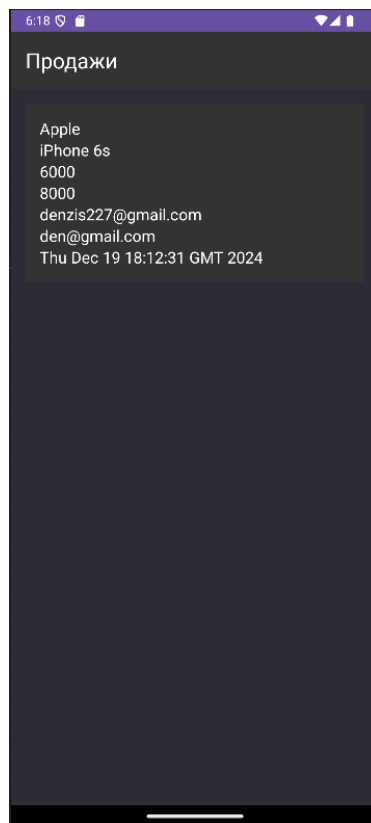


Рисунок 13 – Список всех продаж

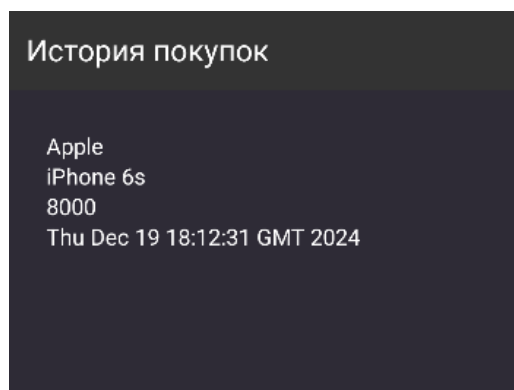


Рисунок 14 – История покупок пользователя

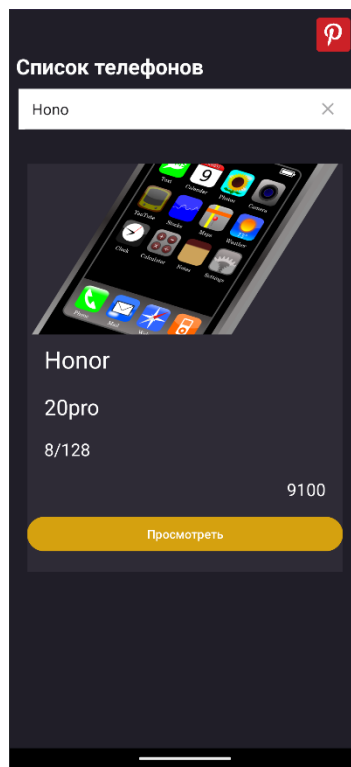


Рисунок 15 –Поиск устройства пользователем

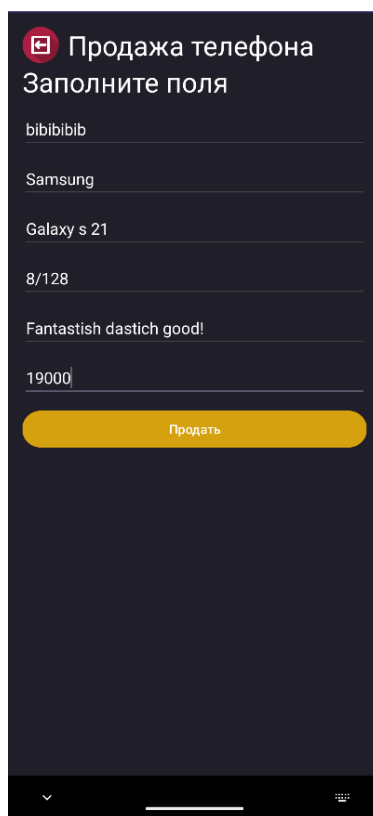


Рисунок 16 – Продажа устройства пользователем