**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

Кафедра ВТиСУ

**КУРСОВАЯ РАБОТА**

по дисциплине «Разработка мобильных приложений»

на тему «Разработка мобильных приложений на платформе Android. Часть 16. Разработка приложения для магазина косметики»

Студент: Сысоева И.В.

Институт: ИИТЭ

Направление: 09.03.03

Группа: ПИ-121

Руководитель: Шутов А.В.

Владимир 2024 г.

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc185109722)

[**ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 4](#_Toc185109723)

[1.1. Роль и место мобильных приложений в современном обществе 4](#_Toc185109724)

[1.2. Анализ популярных мобильных платформ 5](#_Toc185109725)

[1.3. Современные архитектуры разработки мобильных приложений 6](#_Toc185109726)

[**ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ** 12](#_Toc185109727)

[2.1. Постановка задачи 12](#_Toc185109728)

[2.1.1. Краткая характеристика организации 12](#_Toc185109729)

[2.1.2. Описание состава задач 13](#_Toc185109730)

[2.2. Обоснование выбора программных средств 14](#_Toc185109731)

[2.3. Краткое руководство пользователя 17](#_Toc185109732)

[**ЗАКЛЮЧЕНИЕ** 26](#_Toc185109733)

[**СПИСОК ЛИТЕРАТУРЫ** 27](#_Toc185109734)

# **ВВЕДЕНИЕ**

В современном мире мобильные приложения играют ключевую роль в упрощении взаимодействия между бизнесом и потребителями. Они обеспечивают доступ к услугам и товарам в любое время и в любом месте, способствуют повышению лояльности клиентов и позволяют компаниям более эффективно продвигать свою продукцию. Сектор розничной торговли, включая магазины косметики, активно использует мобильные технологии для привлечения покупателей и создания удобного пользовательского опыта.

Целью данной курсовой работы является разработка мобильного приложения для магазина косметики. Приложение должно предоставить пользователям возможность просматривать каталог товаров, изучать карточки товаров.

Для достижения цели будут решены следующие задачи: изучение роли мобильных приложений в современном обществе, анализ популярных платформ и архитектур разработки, выбор оптимального подхода к реализации, а также обоснование используемых программных средств и создание краткого руководства пользователя.

Результатом выполнения курсовой работы станет функциональное мобильное приложение для магазина косметики, которое будет способствовать повышению удобства взаимодействия покупателей с магазином, улучшению сервиса и росту продаж.

# **ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## 1.1. Роль и место мобильных приложений в современном обществе

Мобильные приложения стали неотъемлемой частью повседневной жизни современного общества. Они оказывают значительное влияние на различные аспекты жизни, включая общение, работу, образование, развлечение и бизнес. Удобство использования смартфонов и планшетов, поддерживаемых мобильными приложениями, обусловило их широкое распространение и востребованность.

Одной из ключевых ролей мобильных приложений является повышение доступности услуг и информации. Приложения позволяют пользователям выполнять широкий спектр задач – от заказа товаров и услуг до получения образовательных материалов и управления финансовыми операциями. Это делает их незаменимым инструментом в условиях динамично развивающегося цифрового мира.

С точки зрения бизнеса, мобильные приложения играют важную роль в повышении конкурентоспособности. Компании используют их для улучшения взаимодействия с клиентами, расширения каналов продаж, формирования бренда и внедрения программ лояльности. Магазины косметики, в частности, получают возможность привлекать клиентов через персонализированные рекомендации, акции и скидки, интегрированные в их приложения.

Также мобильные приложения способствуют развитию новых форм коммуникации и социальных взаимодействий. Социальные сети, мессенджеры и платформы для видеоконференций значительно упростили процесс общения и сделали его доступным для миллионов людей по всему миру.

В образовательной и профессиональной среде мобильные приложения помогают автоматизировать процессы обучения, предоставлять доступ к курсам, тренингам и инструментам для организации работы. В здравоохранении они используются для записи к врачу, мониторинга состояния здоровья и предоставления дистанционной медицинской помощи.

Кроме того, мобильные приложения играют значимую роль в личной эффективности и организации времени. Платформы для управления задачами, планирования и отслеживания личных целей помогают пользователям более эффективно распределять свое время и ресурсы.

Таким образом, мобильные приложения стали неотъемлемым элементом цифрового общества, оказывая влияние на широкий спектр сфер жизни. Их адаптация и развитие продолжают задавать новый вектор инновациям и улучшению качества жизни.

## 1.2. Анализ популярных мобильных платформ

Современный рынок мобильных приложений представлен несколькими основными платформами, каждая из которых имеет свои особенности, влияющие на процесс разработки. Наиболее распространенными являются Android и iOS, которые занимают подавляющую часть рынка мобильных устройств.

Android, разработанный компанией Google, является самой популярной мобильной платформой в мире, охватывая более 70% рынка. Широкая доступность устройств на базе Android объясняется разнообразием производителей, предлагающих модели в различных ценовых категориях – от бюджетных до премиальных. Это делает Android особенно популярным в развивающихся странах, таких как Индия, Бразилия и страны Африки.

Главным преимуществом Android является его открытость, которая позволяет разработчикам гибко настраивать приложения и использовать различные подходы к проектированию интерфейса и функциональности. Кроме того, процесс публикации приложений в магазине Google Play отличается относительно низкими затратами и простотой. Однако платформа сталкивается с проблемой фрагментации: разнообразие устройств с разными версиями операционной системы усложняет тестирование и требует дополнительных усилий для обеспечения совместимости приложения.

iOS, напротив, представляет собой закрытую платформу, разработанную Apple. Ее доля рынка составляет около 27%, причем основными пользователями являются жители развитых стран. Преимущество iOS заключается в стабильности экосистемы: ограниченное количество моделей устройств и централизованные обновления операционной системы упрощают процесс тестирования приложений. Кроме того, пользователи iOS характеризуются высокой платежеспособностью, что делает эту платформу привлекательной для разработчиков платных приложений и сервисов с подписками. Тем не менее, для публикации в App Store разработчики сталкиваются с более строгими требованиями и высокими затратами.

Существуют также альтернативные платформы, такие как HarmonyOS от Huawei и KaiOS для бюджетных устройств. Однако их доля рынка остается незначительной по сравнению с Android и iOS.

В рамках данного проекта для разработки мобильного приложения магазина косметики была выбрана платформа Android. Этот выбор обусловлен рядом факторов, среди которых широкая аудитория пользователей, разнообразие устройств и более доступные условия для публикации приложения. Таким образом, ориентируясь на Android, приложение сможет охватить максимально возможное количество пользователей и предложить удобный инструмент для взаимодействия с магазином.

## 1.3. Современные архитектуры разработки мобильных приложений

Архитектура мобильного приложения определяет структуру системы, распределение функций между клиентом и сервером, а также способ взаимодействия с данными. В рамках общей архитектуры часто применяются архитектурные паттерны, которые помогают структурировать логику клиентской части приложения.

Основные типы архитектур мобильных приложений:

1. Клиент-серверная архитектура: Наиболее распространенный подход в разработке мобильных приложений.

Клиент: мобильное приложение на устройстве пользователя, которое обрабатывает интерфейс, отправляет запросы и отображает полученные данные.

Сервер: удаленная система, управляющая базой данных и логикой обработки запросов.

Эта архитектура позволяет централизованно управлять данными, интегрировать внешние сервисы (например, платежные шлюзы, службы доставки) и масштабировать серверную часть для обслуживания большого числа клиентов.

2. Файл-серверная архитектура: В этой архитектуре приложение обращается к серверу для чтения или записи файлов. Этот подход прост в реализации, но ограничен возможностями обработки данных и редко используется в современных мобильных приложениях.

3. Облачная архитектура: Приложения с облачной архитектурой взаимодействуют с данными, размещенными на облачных платформах (например, Google Firebase, AWS). Это обеспечивает доступность данных из любого места, упрощает масштабирование и интеграцию с другими сервисами, но требует постоянного подключения к интернету.

3. Одноранговая (P2P) архитектура: Одноранговые приложения (peer-to-peer) обмениваются данными напрямую между устройствами пользователей без использования центрального сервера. Этот подход подходит для приложений с высокой приватностью или работы в условиях отсутствия стабильного подключения к интернету.

4. Безсерверная архитектура (Serverless): Приложения с безсерверной архитектурой используют облачные функции для обработки запросов. Серверная часть существует только при необходимости выполнения конкретных операций, что снижает затраты на обслуживание инфраструктуры.

5. Монолитная архитектура: Все функциональные компоненты приложения находятся в одном приложении. Подходит для простых проектов, но с увеличением сложности может стать трудной для сопровождения.

6. Микросервисная архитектура: Микросервисная архитектура делит приложение на отдельные, автономные сервисы. Каждый сервис отвечает за конкретную задачу и может быть разработан и обновлен независимо. Это особенно полезно для масштабных систем, но требует сложной инфраструктуры.

Подробнее стоит рассмотреть клиент-серверную архитектуру, используемую в дальнейшей разработке проекта.

Клиент-серверная архитектура – одна из самых распространенных моделей разработки мобильных приложений, особенно для тех, которые требуют постоянного взаимодействия с удаленными базами данных, такими как каталоги товаров или истории заказов. В рамках этой архитектуры приложение делится на две ключевые части: клиентскую и серверную.

Рассмотрим основные элементы клиент-серверной архитектуры

Клиент – это мобильное приложение, установленное на устройстве пользователя, оно отвечает за отображение интерфейса, обработку действий пользователя и отправку запросов на сервер.

Сервер – расположен в удаленном центре обработки данных или облаке.

Обрабатывает запросы клиента, управляет базами данных и возвращает данные или выполняет операции (например, авторизацию, обработку заказов).

Преимущества клиент-серверной архитектуры:

1. Централизация данных: Все данные хранятся на сервере, что обеспечивает их актуальность и упрощает управление.

2. Масштабируемость: Сервер может быть адаптирован для работы с большим количеством клиентов путем увеличения вычислительных мощностей или распределения нагрузки.

3. Безопасность: Хранение данных на сервере позволяет использовать централизованные методы шифрования и защиты информации.

4. Гибкость: Серверная часть может быть обновлена или изменена без необходимости обновлять клиентское приложение.

В выбранном проекте для магазина косметики такая архитектура обеспечивает удобное управление каталогом товаров и обработку заказов пользователей через серверную часть.

Архитектурные паттерны определяют, как структурировать код внутри клиентского приложения. Рассмотрим основные паттерны.

1. MVC (Model-View-Controller)

Model – отвечает за данные и бизнес-логику.

View – отображает данные и обрабатывает пользовательский ввод.

Controller – связывает View и Model, управляя их взаимодействием.

Это базовый и простой в реализации паттерн, но в крупных приложениях View и Controller могут стать слишком тесно связаны.

2. MVVM (Model-View-ViewModel)

MVVM улучшает MVC, добавляя слой ViewModel, который обрабатывает логику пользовательского интерфейса и обеспечивает двустороннюю связь данных между Model и View.

Плюсы: хорошая тестируемость и изоляция логики интерфейса. Часто используется в Android-разработке с библиотеками Jetpack (LiveData, Data Binding).

3. MVP (Model-View-Presenter)

MVP разделяет View и Model, добавляя слой Presenter, который управляет всей бизнес-логикой. View остается "глупой" и отвечает только за отображение данных.

4. VIPER (View-Interactor-Presenter-Entity-Router)

Строгий модульный подход, который делит приложение на отдельные компоненты. Отличается высокой тестируемостью, но сложен в реализации.

Для внутренней организации клиентской части приложения выбран паттерн MVC, который широко используется благодаря своей простоте и четкому разделению обязанностей.

Основные компоненты MVC:

1. Model – представляет данные и бизнес-логику.

Включает взаимодействие с сервером (запросы API), обработку данных (например, фильтрацию, сортировку) и хранение их в локальных структурах (например, базах данных SQLite).

В приложении для магазина косметики это может быть структура, хранящая информацию о товарах, корзине покупателя и заказах.

2. View – отвечает за отображение пользовательского интерфейса и взаимодействие с пользователем.

Это экраны и элементы управления, такие как кнопки, списки и формы ввода. View реагирует на действия пользователя и отображает данные, полученные из Model.

3. Controller – связывает Model и View.

Получает пользовательские запросы из View, обрабатывает их (например, отправляет запрос на сервер через Model) и обновляет View с результатами. В приложении для магазина Controller будет обрабатывать такие действия, как добавление товара в корзину, авторизация пользователя или фильтрация каталога.

Преимущества MVC:

1. Разделение обязанностей: Логика работы приложения разделена между компонентами, что делает код более читаемым и понятным.

2. Легкость модификации: Изменения в одном компоненте минимально влияют на другие, что упрощает разработку и поддержку.

3. Тестируемость: Логика (Model) отделена от интерфейса (View), что позволяет проводить автоматическое тестирование бизнес-логики.

MVC подходит для проекта из-за его простоты, что важно при разработке приложения с ограниченными сроками. В будущем возможно расширение паттерна до более сложных, таких как MVVM, для увеличения тестируемости и масштабируемости.

В результате выбор клиент-серверной архитектуры и паттерна MVC обеспечивает надежность и удобство разработки. Архитектура отвечает за взаимодействие клиента и сервера, а паттерн помогает структурировать логику приложения, что делает его гибким и легким в сопровождении.

# **ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

## 2.1. Постановка задачи

### 2.1.1. Краткая характеристика организации

Название бренда: MineralGlow.

Сфера деятельности: Магазин минеральной косметики.

Краткое описание: Год на рынке. Конкурентное преимущество минеральной косметики заключается в ее натуральности и более мягком воздействии на кожу по сравнению с традиционной косметикой. Она безопасна для людей с аллергией и чувствительной кожей, имеет солнцезащитный фактор, не тестируется на животных, экологична.

Перечень продаваемых товаров: Минеральная основа, румяна, хайлайтер, скульптор, палетка теней, тени для бровей, пигменты, гель для бровей, Масло для губ.

Каналы продвижения: соцсети, поисковое продвижение, реклама в журналах, участие в ярмарках и выставках.

Цели и задачи, которые планируется достичь посредством разрабатываемого сайта: Привлечение новых клиентов, продажи через сайт, продвижение новых товаров, информирование аудитории, укрепление делового имиджа.

Целевая аудитория:

Пол: женский

Возраст: от 18 до 40 лет

Социальный статус: средний

Хобби: интерес к здоровому образу жизни, спорту, путешествиям, экологически чистой продукции.

Другие характеристики, которые могут быть важными для целевой аудитории минеральной косметики:

• Интерес к натуральным и органическим продуктам.

• Проблемы с кожей, такие как чувствительность, акне, пигментация и т.д.

• Желание использовать косметику, которая не содержит вредных химических веществ.

• Желание использовать косметику, которая не тестируется на животных.

• Желание использовать косметику, которая помогает сохранять здоровье кожи и предотвращать ее старение.

### 2.1.2. Описание состава задач

Задачи, автоматизируемые приложением:

1. Авторизация пользователей.

Приложение обеспечивает возможность регистрации и авторизации клиентов. Эта функция автоматизирует процесс идентификации клиентов, делает работу с приложением персонализированной и удобной.

2. Управление каталогом товаров.

Приложение предоставляет пользователям доступ к интерактивному каталогу продукции. Система автоматизирует задачу отображения информации о товарах, включая их наименование, фотографии, текущую цену. Это устраняет необходимость в ручном обновлении данных и облегчает управление ассортиментом.

3. Карточка товара.

Приложение предоставляет детализированную информацию о каждом товаре через карточку товара. В ней отображаются такие данные, как описание продукта, характеристики, цена. Это автоматизирует предоставление информации о товарах, делая процесс выбора более осознанным.

4. Формирование корзины покупок.

Приложение автоматизирует процесс формирования корзины, позволяя клиентам добавлять выбранные товары и видеть итоговую сумму заказа. Это значительно улучшает пользовательский опыт.

5. Административная панель.

Администратор получит возможность управлять каталогом товаров, добавлять новые и удалять устаревшие товары.

Реализуемые функции – авторизация, каталог товаров, карточка товара, корзина и административная панель – автоматизируют ключевые задачи, с которыми сталкиваются клиенты и сотрудники магазина. Это не только улучшает удобство использования приложения, но и оптимизирует бизнес-процессы, связанные с продажей продукции и взаимодействием с покупателями.

## 2.2. Обоснование выбора программных средств

При разработке мобильного приложения для магазина косметики использовались различные программные средства, каждое из которых было выбрано на основе их функциональных возможностей, популярности и соответствия требованиям проекта.

Система управления базами данных:

Для управления данными приложения выбрана MySQL, одна из самых популярных реляционных систем управления базами данных. MySQL используется как для небольших, так и для крупных проектов благодаря своей производительности и надежности.

MySQL поддерживает мощный язык запросов SQL, что позволяет эффективно работать с большими объемами данных. Преимуществами MySQL являются:

* высокая скорость выполнения операций, даже при больших объемах данных;
* возможность интеграции с различными языками программирования, включая PHP;
* поддержка репликации и масштабирования баз данных;
* наличие бесплатной версии.

Однако MySQL имеет и свои недостатки, такие как ограниченные возможности работы с нетипизированными данными и отсутствие встроенной поддержки полнотекстового поиска для некоторых типов запросов. Тем не менее, для данного проекта MySQL был выбран благодаря его стабильности, интеграции с PHP и широкому распространению.

Средства разработки клиентской части:

Для создания клиентской части приложения использовалась Android Studio, официальная среда разработки для платформы Android. Android Studio предоставляет полный набор инструментов для разработки, включая визуальный редактор интерфейса, инструменты отладки и профилировки, а также эмулятор Android.

Преимущества Android Studio включают:

* глубокую интеграцию с Android SDK, что позволяет использовать все возможности операционной системы;
* поддержку современных стандартов разработки, включая инструменты для работы с дизайном и оптимизации производительности;
* широкое сообщество пользователей и обширную документацию, которые упрощают процесс обучения и решения возникающих проблем.

К недостаткам можно отнести высокие системные требования, что может создавать трудности при использовании менее производительных компьютеров. Несмотря на это, Android Studio была выбрана как наиболее подходящая среда для разработки приложений на платформе Android.

Клиентская часть была реализована на языке Java, который является стандартом для разработки Android-приложений. Java обеспечивает стабильность, обширную базу знаний и поддержку всех необходимых библиотек. Хотя Java иногда критикуется за более сложный синтаксис по сравнению с Kotlin, для данного проекта он был выбран как проверенный и широко используемый инструмент.

Средства разработки серверной части:

Для разработки серверной части приложения был использован редактор Visual Studio Code (VS Code). Этот инструмент отличается легкостью и гибкостью, поддерживая расширения для работы с разными языками программирования.

Преимущества VS Code заключаются в его высокой скорости работы, удобстве написания кода благодаря подсветке синтаксиса и поддержке IntelliSense, а также возможности кастомизации с помощью плагинов. Недостатком является отсутствие встроенных мощных инструментов для работы с большими проектами, но это компенсируется подключением необходимых расширений.

Серверная логика реализована на языке PHP, который широко используется для создания серверных приложений. PHP был выбран благодаря его простоте, большому количеству готовых решений и хорошей интеграции с MySQL. Среди преимуществ PHP можно выделить его популярность, низкий порог входа и большое количество доступных библиотек. Недостатками являются менее строгая типизация и необходимость ручной оптимизации при больших нагрузках.

Выбор программных средств, включая MySQL, Android Studio, Java, Visual Studio Code и PHP, был сделан на основе их функциональности и способности эффективно решать задачи проекта. Эти инструменты позволяют обеспечить надежность, производительность и удобство разработки мобильного приложения для магазина косметики.

## 2.3. Краткое руководство пользователя

В данном разделе представлено краткое руководство пользователя, описывающее основные элементы интерфейса и функциональные возможности мобильного приложения для магазина косметики. Руководство поможет пользователям быстро освоить работу с приложением и эффективно использовать его функционал.

При входе в приложение открывается форма авторизации (рис.1), для входа в приложение нужно ввести логин, пароль и нажать кнопку «Войти». Если такой пользователь отсутствует, то приложение не откроется, тогда можно нажать кнопку «Зарегистрироваться».

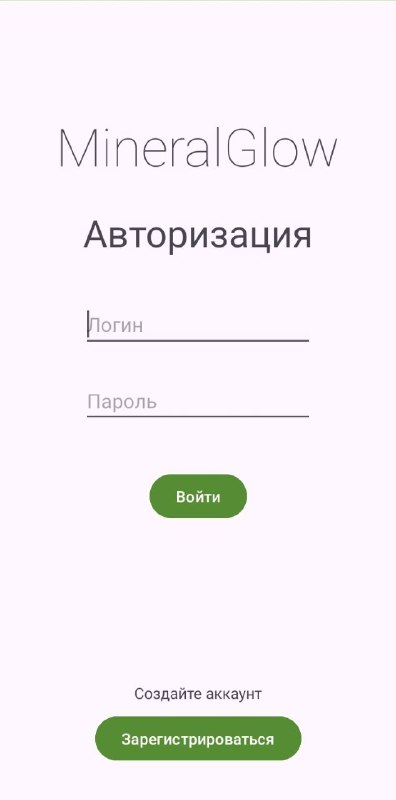


Рис.1 – Форма авторизации.

После нажатия кнопки «Зарегистрироваться» открывается форма регистрации (Рис.2), там пользователю необходимо придумать логин, пароль и нажать кнопку для регистрации, после этого можно войти под этим логином и паролем.

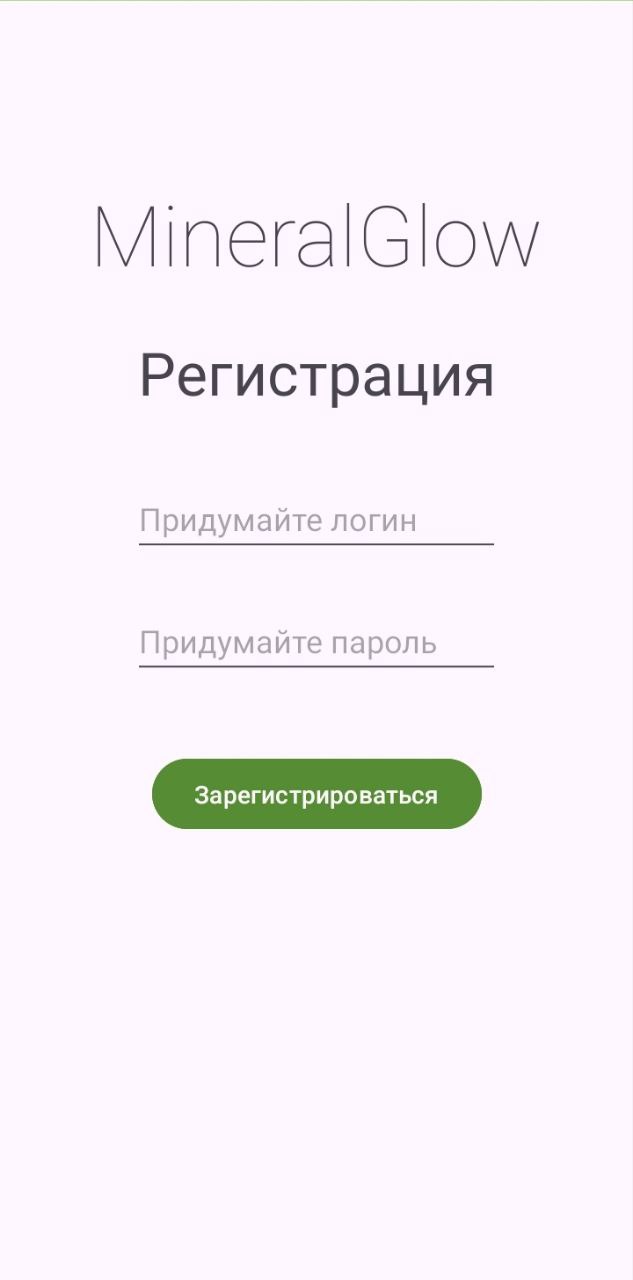


Рис.2 – Форма регистрации.

После входа открывается каталог товаров (Рис.3), он организован в виде сетки, что позволяет удобно просматривать позиции. Можно увидеть фото товаров, название и цену, также есть кнопка добавления в корзину. В низу находится меню, слева расположена кнопка личного кабинета, по центру главная (каталог) и справа корзина.

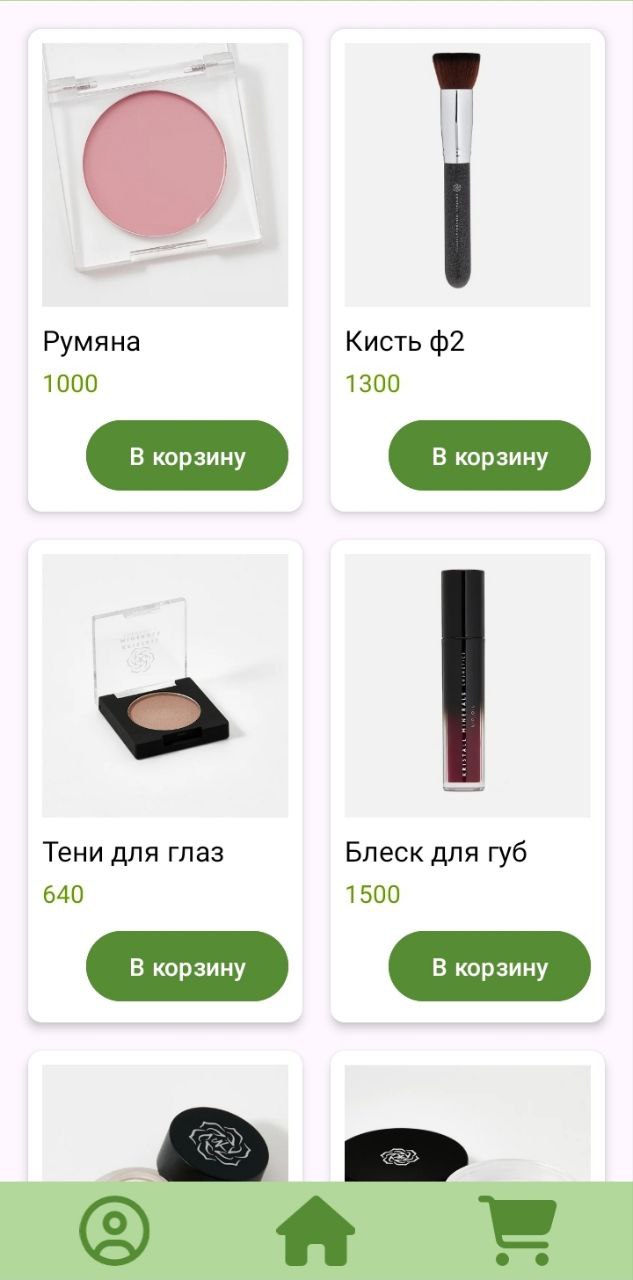


Рис.3 – Каталог товаров.

При выборе товара открывается карточка товара (рис.4), где можно ознакомиться с подробной информацией и добавить товар в корзину. В карточке можно посмотреть такую информацию как название товара, цена и описание.



Рис.4 – Карточка товара.

В разделе корзина (Рис.5) отображаются все выбранные товары и внизу расположена кнопка «Заказать».



Рис.5 – Корзина.

В личном кабинете можно выйти из своего профиля (Рис.6).



Рис.6 – Личный кабинет.

С аккаунта администратора в личном кабинете (рис.7) есть дополнительная кнопка «Добавить товар».

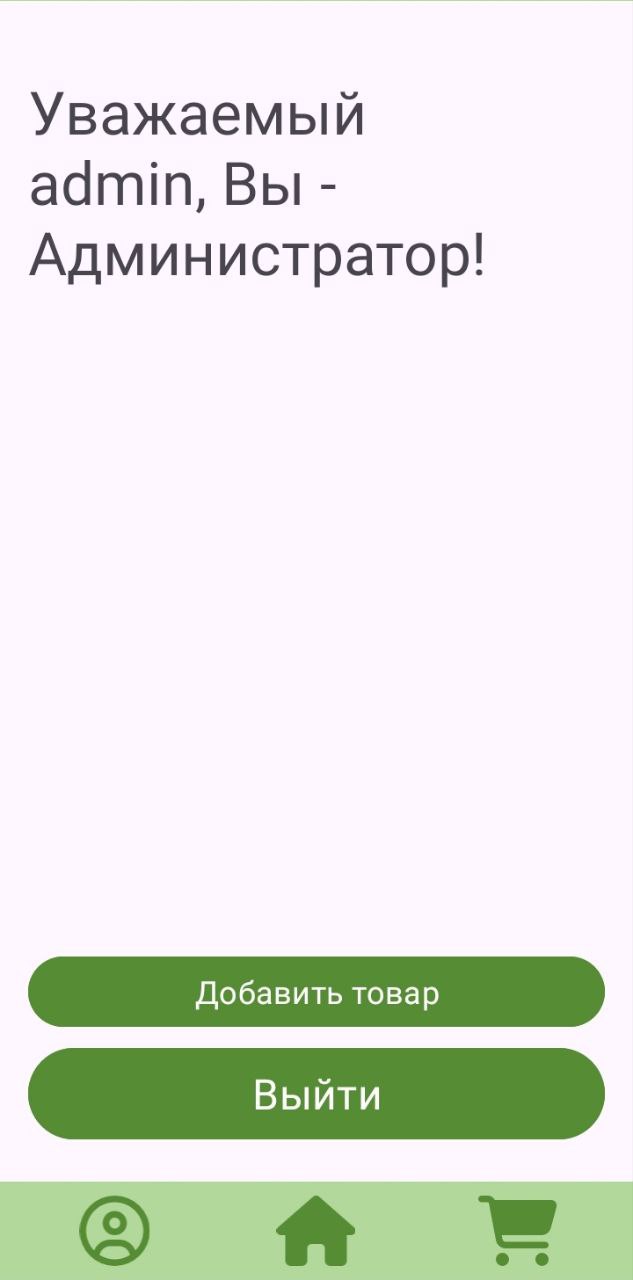


Рис.7 – Личный кабинет admin.

При добавлении товара открывается форма (Рис.8), в которой нужно добавить название товара, цену, описание и фото, после нажатия кнопки «Добавить товар» от появится в каталоге.

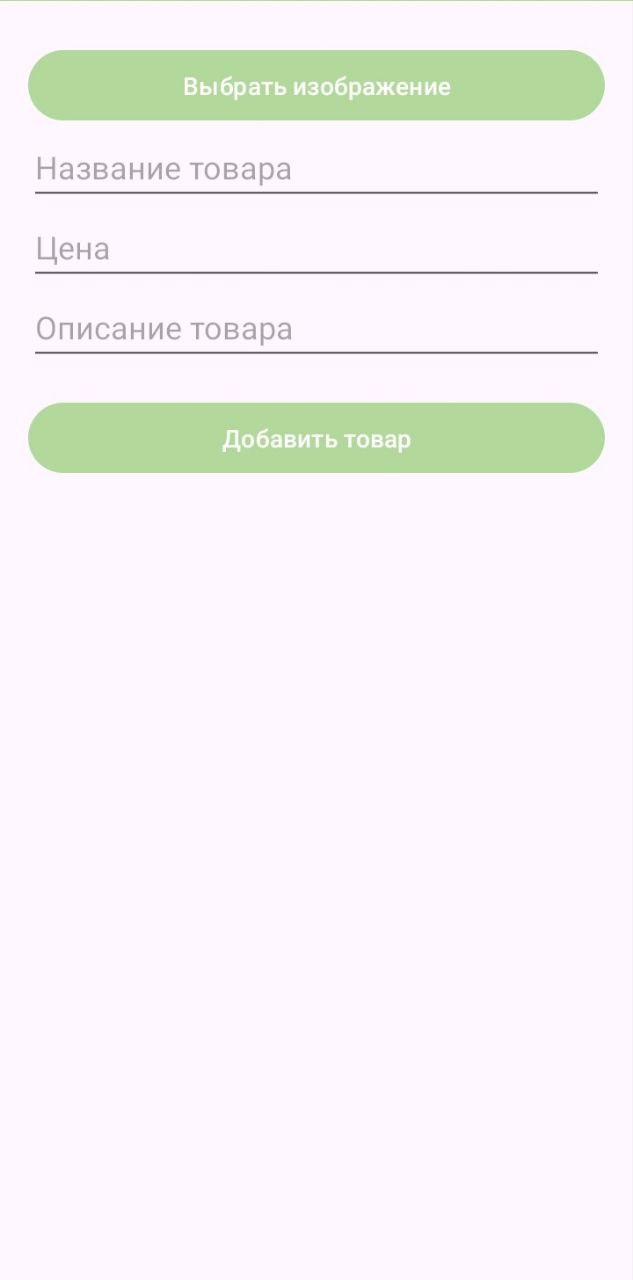


Рис.8 – Форма добавления товара.

В карточке товара у администратора (рис.9) тоже появляется дополнительная кнопка для удаления товара.

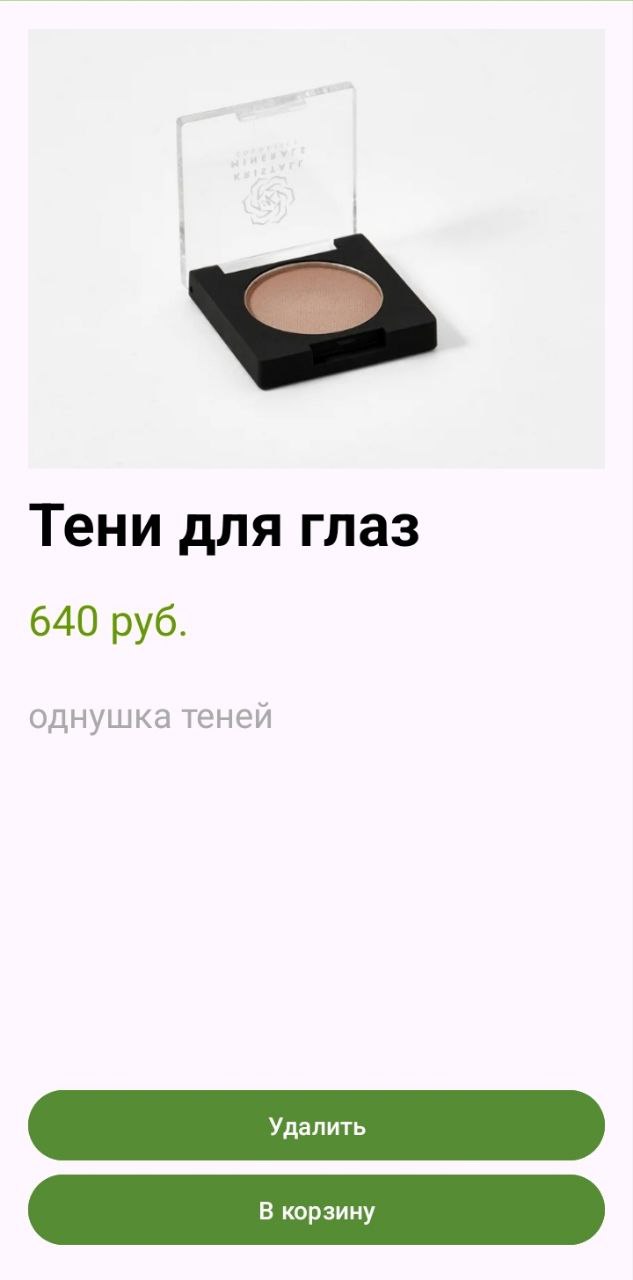


Рис.9 – Карточка товара admin.

Приложение разработано с учетом принципов удобства использования и интуитивности интерфейса, что делает его доступным для пользователей любого уровня подготовки.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы было разработано мобильное приложение для магазина косметики, целью которого стало упрощение взаимодействия с клиентами и повышение эффективности бизнес-процессов. В процессе разработки были проанализированы современные архитектуры, платформы и инструменты создания мобильных приложений, что позволило обосновать выбор технологий и средств, наиболее подходящих для реализации поставленных задач.

В теоретической части курсовой работы рассмотрены ключевые особенности мобильных приложений, их роль и место в современном обществе. Был проведен анализ популярных мобильных платформ. Также изучены архитектурные подходы.

Практическая часть работы была посвящена реализации функционала, включающего авторизацию пользователей, управление каталогом товаров, просмотр карточек товаров и формирование заказов через корзину.

Результаты работы демонстрируют, что выбранные архитектурные подходы, технологии и инструменты полностью удовлетворяют требованиям проекта. Разработанное приложение обладает интуитивно понятным интерфейсом, функциональностью, необходимой для работы магазина, и возможностью дальнейшего расширения.

Таким образом, цель работы была достигнута: создано мобильное приложение, которое повышает удобство обслуживания клиентов магазина косметики и способствует оптимизации его бизнес-процессов. Полученные знания и навыки могут быть применены для разработки аналогичных приложений и углубления опыта в области мобильной разработки.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Рубцов, А. Разработка мобильных приложений для Android [Текст] / А. Рубцов. – М.: БХВ-Петербург, 2020. – 448 с.
2. Блинов, Л. Язык программирования Java. Учебное пособие [Текст] / Л. Блинов, А. Романчик. – Минск: Новое знание, 2019. – 720 с.
3. Кушнир, А. Разработка клиент-серверных приложений [Текст] / А. Кушнир. – М.: Диалог, 2020. – 384 с.
4. Эндрюс, К. Основы Android-разработки [Текст] / К. Эндрюс. – СПб.: Питер, 2021. – 496 с.