**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 18. Разработка приложения для магазина машин»

Студент: Черноморченко К.О.

Институт: ИИТЭ

Направление: 09.03.03

Группа: ПИ-121

Руководитель: Шутов А.В.

Владимир 2024 г.

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc185173268)

[**ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 4](#_Toc185173269)

[1.1. Роль и место мобильных приложений в современном обществе 4](#_Toc185173270)

[1.2. Анализ популярных мобильных платформ 6](#_Toc185173271)

[1.3. Современные архитектуры разработки мобильных приложений 8](#_Toc185173272)

[**ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ** 12](#_Toc185173273)

[2.1. Постановка задачи 12](#_Toc185173274)

[2.1.1. Краткая характеристика организации 12](#_Toc185173275)

[2.1.2. Описание состава задач 12](#_Toc185173276)

[2.2. Обоснование выбора программных средств 13](#_Toc185173277)

[2.3. Краткое руководство пользователя 15](#_Toc185173278)

[**ЗАКЛЮЧЕНИЕ** 22](#_Toc185173279)

[**СПИСОК ЛИТЕРАТУРЫ** 23](#_Toc185173280)

# **ВВЕДЕНИЕ**

Современные технологии стремительно развиваются, и мобильные устройства стали неотъемлемой частью повседневной жизни людей. Мобильные приложения играют важную роль в упрощении различных процессов, таких как покупки, управление финансами, обучение и многое другое. В условиях растущего спроса на удобство и доступность информации через смартфоны и планшеты, разработка мобильных приложений становится актуальной задачей для бизнеса.

Целью данной курсовой работы является создание мобильного приложения для магазина машин, которое позволит пользователям легко находить нужную информацию о транспортных средствах и узнавать цены.

Для достижения поставленной цели необходимо решить следующие задачи:

* Изучить современные подходы к разработке мобильных приложений.
* Провести сравнительный анализ популярных мобильных платформ.
* Определить архитектуру и инструменты для создания мобильного приложения.
* Разработать функциональную спецификацию будущего приложения.
* Подготовить краткое руководство пользователя для разработанного приложения.

Таким образом, данная работа охватывает весь цикл разработки мобильного приложения – от анализа требований до создания готового продукта.

# **ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

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

Мобильные приложения прочно вошли в повседневную жизнь современного общества благодаря широкому распространению смартфонов и планшетных устройств. Они представляют собой специализированные программы, предназначенные для выполнения конкретных функций на мобильных устройствах. С каждым годом количество пользователей мобильных приложений растет, что делает их важным инструментом для взаимодействия между людьми, бизнесом и государственными структурами.

Одним из ключевых преимуществ мобильных приложений является их доступность и удобство использования. Пользователи могут получить доступ к необходимой информации или услугам в любое время и в любом месте при наличии интернет-соединения. Это позволяет значительно экономить время и ресурсы, особенно в сравнении с традиционными методами получения услуг.

Кроме того, мобильные приложения предоставляют возможность персонализации контента под нужды конкретного пользователя. Например, в приложениях для электронной коммерции пользователи могут получать рекомендации товаров на основе своих предпочтений и истории покупок. Также они позволяют использовать push-уведомления для информирования об акциях, скидках и других важных событиях.

Мобильные приложения находят применение практически во всех сферах жизнедеятельности:

1. Электронная коммерция: Приложения для онлайн-магазинов, такие как Ozon, Wildberries, AliExpress, позволяют покупателям совершать покупки без необходимости посещения физических магазинов.

2. Образование: Образовательные платформы, такие как Coursera, Udemy, предоставляют доступ к учебным материалам и курсам, позволяя обучаться дистанционно.

3. Финансы: Банковские приложения, такие как Сбербанк Онлайн, Тинькофф, дают возможность управлять счетами, оплачивать услуги и переводить деньги без визита в банк.

4. Развлечения: Социальные сети, стриминговые сервисы (например, Netflix, Spotify), игры и другие развлекательные приложения занимают значительную долю времени пользователей.

5. Транспорт и логистика: Приложения для заказа такси (Yandeх.Taxi, Uber), отслеживания посылок и доставки еды (Delivery Club, Яндекс.Еда) делают перемещение и доставку удобными и быстрыми.

6. Здоровье и фитнес: Приложения для мониторинга физической активности (Google Fit, Strava), контроля питания и здоровья помогают людям следить за своим состоянием и улучшать качество жизни.

7. Государственные услуги: Порталы государственных услуг, такие как Госуслуги, позволяют гражданам получать различные документы и справки, подавать заявления и отслеживать статус обращений.

Мобильные приложения оказывают значительное влияние на развитие бизнеса. Компании, использующие мобильные приложения, получают ряд преимуществ:

* Увеличение клиентской базы: Благодаря доступности приложений на глобальном рынке, компании могут привлекать новых клиентов из разных регионов и стран.
* Повышение лояльности клиентов: Удобство использования приложений и персонализированный подход способствуют удержанию клиентов и повышению их удовлетворенности.
* Оптимизация бизнес-процессов: Автоматизация рутинных операций, таких как обработка заказов, учет запасов и управление персоналом, позволяет сократить затраты и повысить эффективность работы.
* Анализ данных: Сбор и анализ пользовательских данных помогает компаниям лучше понимать потребности своей аудитории и адаптировать свои продукты и услуги.

Таким образом, мобильные приложения продолжают играть ключевую роль в современной жизни, обеспечивая удобный доступ к различным сервисам и улучшая взаимодействие между пользователями и организациями.

## 1.2. Анализ популярных мобильных платформ

Современный рынок мобильных приложений представлен двумя основными платформами: Android и iOS. Каждая из них имеет свои особенности, преимущества и недостатки, которые следует учитывать при выборе платформы для разработки мобильного приложения.

Android – это открытая операционная система, разработанная компанией Google. Она используется на большинстве смартфонов и планшетов по всему миру. Основные характеристики Android включают:

1. Широкая аудитория: Android занимает около 70% мирового рынка мобильных устройств, что обеспечивает огромный охват потенциальных пользователей.

2. Открытый исходный код: Разработчики имеют доступ к исходному коду системы, что позволяет создавать кастомизированные решения и адаптировать их под конкретные нужды.

3. Многообразие устройств: Android поддерживается множеством производителей, включая Samsung, Huawei, Xiaomi и др., что дает пользователям широкий выбор устройств с разными характеристиками и ценами.

4. Легкость публикации: Процесс публикации приложений в Google Play Market относительно прост и не требует строгого соблюдения правил, как в случае с App Store.

Недостатки Android включают фрагментацию версий операционной системы и разнообразие аппаратного обеспечения, что может усложнить тестирование и поддержку приложений.

iOS – это закрытая операционная система, разработанная Apple Inc. для устройств компании, таких как iPhone и iPad. Основные характеристики iOS включают:

1. Высокий доход пользователей: Владельцы устройств на базе iOS обычно обладают высоким уровнем дохода, что делает эту платформу привлекательной для разработчиков коммерческих приложений.

2. Единственный производитель: Все устройства работают на одной версии операционной системы, что облегчает разработку и тестирование приложений.

3. Строгая модерация: Приложения проходят тщательную проверку перед публикацией в App Store, что гарантирует высокое качество и безопасность приложений.

4. Интеграция с экосистемой Apple: Приложения могут легко интегрироваться с другими продуктами Apple, такими как MacBook, Apple Watch и HomePod.

Недостатками iOS являются высокая стоимость разработки и строгие требования к дизайну и функционалу приложений, установленные Apple.

Для данного проекта было принято решение разработать мобильное приложение на платформе Android. Это обусловлено следующими причинами:

1. Большая аудитория: Android занимает лидирующие позиции на мировом рынке мобильных устройств, что обеспечит широкое распространение нашего приложения.

2. Доступность разработки: Открытый исходный код и наличие множества бесплатных инструментов для разработки позволят снизить затраты на создание приложения.

3. Гибкость и масштабируемость: Возможность адаптации приложения под разные устройства и версии операционной системы даст нам больше возможностей для дальнейшего развития проекта.

Таким образом, платформа Android предоставляет оптимальные условия для реализации проекта, учитывая его специфику и целевую аудиторию.

## 1.3. Современные архитектуры разработки мобильных приложений

Архитектура мобильного приложения определяет структуру и организацию программного кода, взаимодействие компонентов и способы обмена данными между ними. От правильного выбора архитектуры зависят ключевые показатели производительности, масштабируемости и надежности приложения. В данном разделе рассмотрим наиболее распространенные архитектурные подходы к разработке мобильных приложений.

Традиционной считается монолитная архитектура, при которой все компоненты приложения находятся в одном блоке кода. Этот метод прост в реализации и тестировании, поэтому он отлично подходит для небольших и простых приложений. Однако по мере роста приложения монолит становится сложным в поддержке и расширении.

Альтернативой является микросервисная архитектура, которая подразумевает разделение приложения на множество независимых сервисов, каждый из которых выполняет свою конкретную функцию. Сервисы взаимодействуют друг с другом посредством API. Данный подход позволяет легко добавлять новые функции и изменять существующие, повышая тем самым масштабируемость и отказоустойчивость приложения.

Еще одним широко используемым подходом является клиент-серверная архитектура, при которой приложение состоит из двух частей: клиента (мобильное устройство) и сервера (удалённого сервера). Клиент отправляет запросы на сервер, получает данные и отображает их пользователю. Сервер, в свою очередь, обрабатывает запросы, хранит данные и управляет бизнес-логикой. Данная архитектура прекрасно подходит для приложений, требующих синхронизации данных между различными устройствами и централизованного управления контентом.

Также существует группа архитектурных паттернов, предназначенных для разделения логики представления и бизнес-логики в приложении: MVC (Model-View-Controller), MVP (Model-View-Presenter) и MVVM (Model-View-ViewModel). Эти паттерны обеспечивают модульность, повторное использование кода и облегчают тестирование отдельных компонентов.

Другой интересный подход – Progressive Web Apps (PWA), которые объединяют функциональность веб-сайтов и мобильных приложений, предоставляя пользователю возможность установки приложения на домашний экран и работы оффлайн. PWA используют современные веб-технологии, такие как Service Workers и Web App Manifest.

Cross-platform development – это подход, позволяющий разрабатывать одно приложение для нескольких платформ одновременно. Популярные фреймворки для кросс-платформенной разработки включают React Native, Flutter и Xamarin. Эти инструменты позволяют писать код один раз и компилировать его для разных платформ, что существенно экономит время и ресурсы.

Наконец, serverless architecture – это архитектура, при которой разработчики сосредотачиваются исключительно на написании кода, а инфраструктура и управление серверами предоставляются облачными провайдерами. Примером такой архитектуры является использование AWS Lambda или Google Cloud Functions.

Для данного проекта была выбрана клиент-серверная архитектура. Этот подход подходит для мобильного приложения магазина автомобилей, так как он позволяет централизованно хранить и обрабатывать данные о товарах, заказах и пользователях. Рассмотрим этот подход более детально.

Причины выбора клиент-серверной архитектуры:

1. Централизованное хранение данных: Все данные о товарах, заказах и пользователях хранятся на сервере, исключая необходимость дублирования информации на каждом устройстве.

2. Синхронизация данных: Пользователи могут работать с актуальными данными независимо от устройства, на котором они авторизованы.

3. Масштабируемость: При увеличении нагрузки на систему можно легко добавить дополнительные серверы или улучшить инфраструктуру без изменения клиентского приложения.

4. Безопасность: Сервер может обеспечить защиту данных и контроль доступа, снижая риски утечки информации.

Основные элементы клиент-серверной архитектуры:

1. Клиентское приложение: Мобильное приложение, установленное на устройстве пользователя. Оно отправляет запросы на сервер и отображает полученные данные.

2. Сервер: Удалённое хранилище данных и логика обработки запросов. Сервер может быть реализован на различных языках программирования и технологиях, таких как Node.js, Python, Java и др.

3. API: Интерфейс взаимодействия между клиентом и сервером. Обычно реализуется с использованием RESTful или GraphQL стандартов.

Для реализации внутренней структуры клиентского приложения был выбран паттерн MVC. Он позволяет разделить логику приложения на три компонента:

1. Модель (Model): Представляет данные и бизнес-логику. Модель отвечает за обработку данных и взаимодействие с базой данных.

2. Вид (View): Отвечает за визуальное представление данных. Вид получает данные от контроллера и отображает их пользователю.

3. Контроллер (Controller): Координирует взаимодействие между моделью и видом. Контроллер принимает входные данные от пользователя, передает их модели для обработки и возвращает результат виду для отображения.

Преимущества использования MVC:

1. Модульность: Каждый компонент может быть изменён или заменён без влияния на остальные части приложения.

2. Повторное использование кода: Модели и контроллеры могут быть использованы в разных видах, уменьшая объём дублирующего кода.

3. Простота тестирования: Компоненты могут быть протестированы отдельно, что упрощает выявление ошибок и багов.

Использование клиент-серверной архитектуры и паттерна MVC в проекте позволит создать надёжное, масштабируемое и удобное в использовании мобильное приложение для магазина автомобилей.

# **ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

## 2.1. Постановка задачи

### 2.1.1. Краткая характеристика организации

Название бренда: CarDrive.

Сфера деятельности: Доска объявлений по продаже автомобилей.

Краткое описание: Год на рынке. Специально обученный персонал производит покупку различных б/у авто. Далее приводит авто в надлежащий вид из-за чего стоимость авто несколько выше вторичного рынка, но покупатель получает уже готовый к эксплуатации авто с гарантией.

Перечень продаваемых товаров: Б/у автотранспорт.

Каналы продвижения: Размещение таргетированной рекламы в поисковиках, Заказ рекламы на различных площадках, связанных с целевой аудиторией. (Соцсети и видеохостинги)

Цели и задачи, которые планируется достичь посредством разрабатываемого сайта: Продажи через сайт, привлечение новых клиентов, информирование аудитории, укрепление делового имиджа.

Целевая аудитория:

Пол: м/ж

Возраст: от 18 лет

Социальный статус: средний, средневысокий.

### 2.1.2. Описание состава задач

Для успешного функционирования мобильного приложения необходимо реализовать следующие задачи:

Авторизация пользователей: Будут разработаны механизмы регистрации и входа в систему. Пользователям будет предложено создать профиль.

Каталог товаров: В приложении будет доступен каталог автомобилей, где пользователи смогут просматривать объявления, размещаемые другими пользователями. Каталог будет организован таким образом, чтобы пользователи могли удобно ориентироваться и находить интересующие их предложения.

Карточка товара: Каждое объявление будет содержать подробную информацию о конкретном автомобиле, включая фотографии и технические характеристики.

Реализовав эти задачи, мы создадим эффективный инструмент для поиска и бронирования автомобилей, который упростит процесс выбора и покупки для конечных пользователей.

## 2.2. Обоснование выбора программных средств

Выбор программных средств для разработки мобильного приложения основывается на ряде критериев, таких как доступность инструментов, их совместимость с платформой, опыт команды разработчиков и требования к проекту. Ниже представлено подробное обоснование выбора инструментов для клиентской и серверной частей приложения.

Для разработки клиентской части мобильного приложения были выбраны следующие инструменты:

Android Studio – официальная интегрированная среда разработки (IDE) для создания приложений под Android. Она предоставляет богатый набор инструментов для написания, тестирования и отладки кода, а также поддерживает интеграцию с различными библиотеками и сервисами. Android Studio включает в себя эмуляторы для тестирования приложений на различных версиях Android, что позволяет убедиться в корректной работе приложения на широком спектре устройств. Кроме того, Android Studio тесно интегрирован с системой управления версиями Git, что упрощает командную работу над проектом.

Java – основной язык программирования для разработки приложений под Android. Он обладает высокой степенью безопасности, стабильностью и широкой поддержкой сообщества разработчиков. Использование Java позволяет создавать надежные и производительные приложения, которые легко интегрируются с существующими системами. Язык Java предоставляет мощные библиотеки для работы с сетью, графикой и пользовательским интерфейсом, что делает его идеальным выбором для разработки сложных мобильных приложений. Также Java активно развивается и поддерживается корпорацией Oracle, что гарантирует долгосрочную поддержку и обновления.

Эти инструменты были выбраны, так как они предоставляют полный набор необходимых функций для разработки качественного мобильного приложения под Android. Android Studio обеспечивает удобную среду для работы, а Java – проверенный временем язык, который идеально подходит для сложных мобильных приложений.

Для разработки серверной части приложения использовались следующие средства:

Visual Studio Code (VS Code) – легкий и мощный редактор кода, который поддерживает множество языков программирования, включая PHP. Он удобен для быстрой разработки и отладки серверного кода благодаря встроенным инструментам и обширной библиотеке расширений. VS Code позволяет настраивать рабочие пространства под индивидуальные потребности, а также интегрируется с различными системами контроля версий и инструментами CI/CD, что ускоряет процесс разработки и деплоя.

PHP – популярный язык программирования для веб-разработки, который известен своей простотой и гибкостью. PHP позволяет быстро создавать серверные скрипты, которые взаимодействуют с базами данных и внешними сервисами. Он хорошо интегрируется с MySQL и другими СУБД. PHP также широко используется в мире веб-разработки, что означает наличие большого количества готовых библиотек и фреймворков, таких как Laravel, Symfony и Yii, которые ускоряют разработку и повышают стабильность приложения.

MySQL – одна из самых популярных реляционных систем управления базами данных (СУБД). Она проста в установке и управлении, обладает высокой скоростью работы и надежностью. MySQL идеально подходит для хранения и обработки больших объемов данных, что критически важно для приложений с большим количеством пользователей и транзакций. MySQL также поддерживает репликацию и кластеризацию, что повышает устойчивость и масштабируемость системы.

Эти инструменты были выбраны из-за их совместимости и легкости в настройке. Visual Studio Code предоставляет удобные возможности для работы с PHP-кодом, а сам PHP позволяет быстро развернуть серверную часть приложения. MySQL обеспечивает надежную и быструю работу с данными, что крайне важно для стабильности и производительности приложения.

Выбранные программные средства соответствуют требованиям проекта и обеспечивают высокий уровень производительности, безопасности и удобства в разработке.

## 2.3. Краткое руководство пользователя

При запуске приложения появляется форма авторизации представленная на рис.1, в которой нужно указать свой логин и пароли, после нажать на кнопку войти, если такой пользователь зарегистрирован, то пользователь войдет в приложение. Если аккаунт отсутствует, то ниже находится кнопка регистрации.

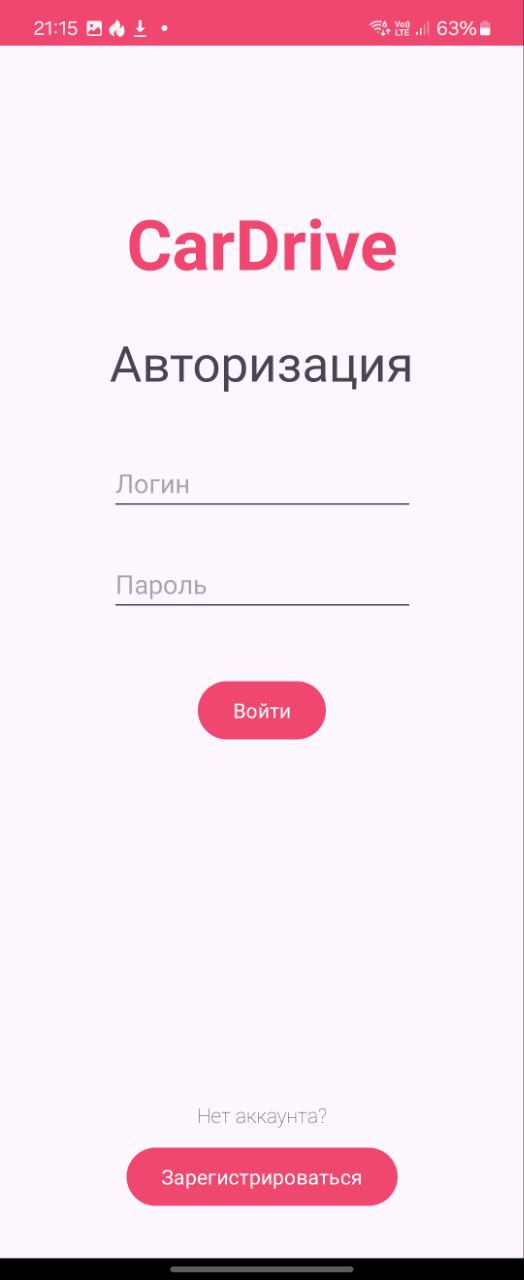


Рис.1 – Форма авторизации.

На рис.2 показана форма регистрации, пользователю необходим придумать логин, пароль и указать номер телефона, после чего нужно нажать кнопку зарегистрироваться, и пользователь попадет в приложение.

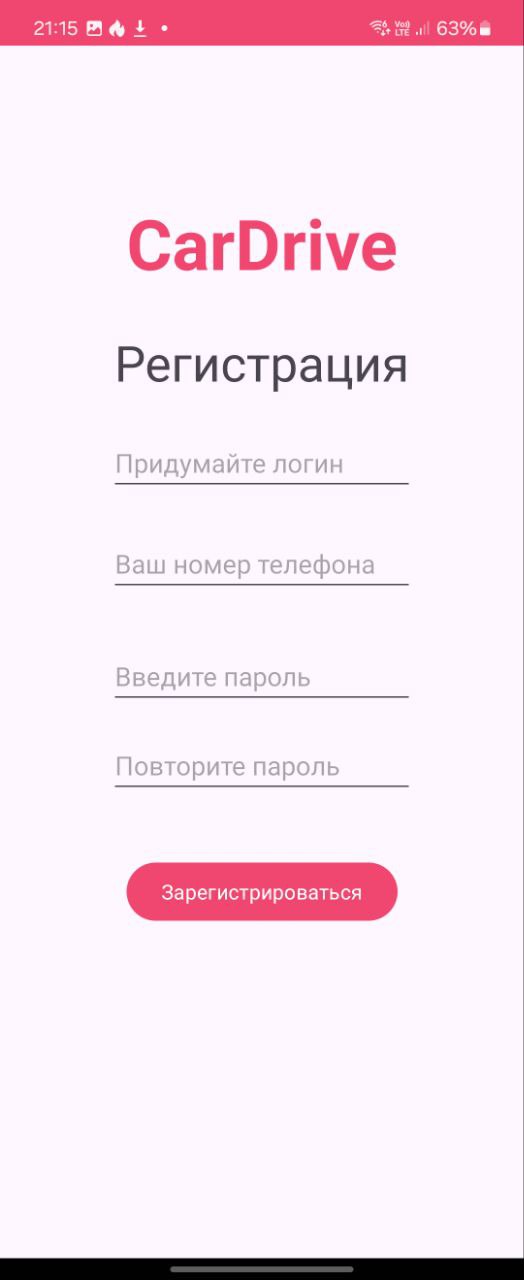


Рис.2 – Форма регистрации.

После входа в аккаунт открывается главная страница (рис.3), на которой показаны все объявления, можно увидеть фотографию машины, ее название и цену. Сверху можно увидеть кнопку добавления нового объявления и кнопку ля перехода в личный кабинет.

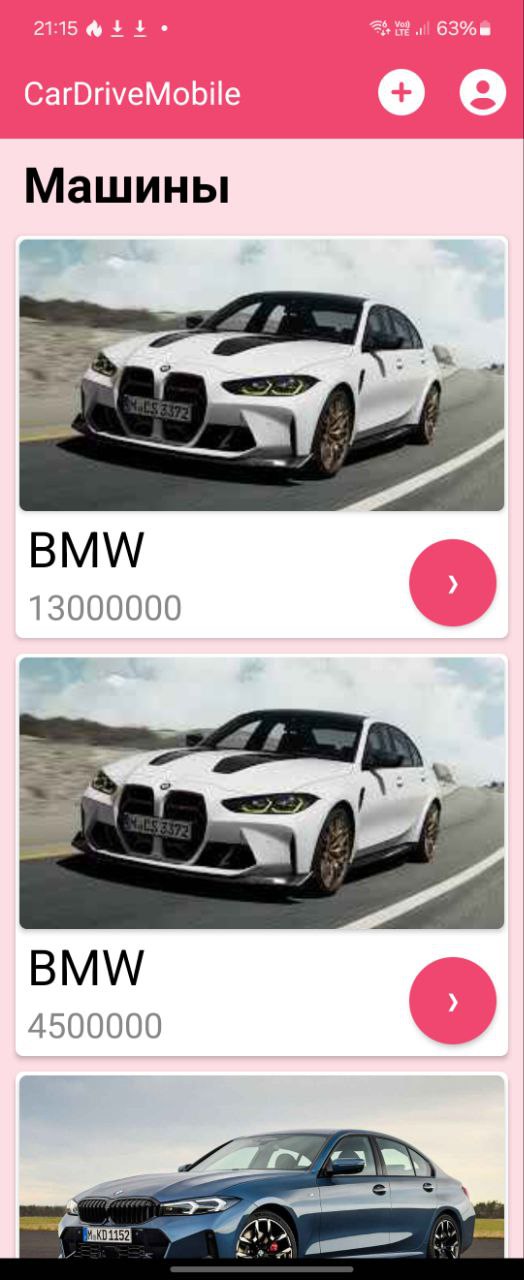


Рис.3 – Главная страница.

При нажатии на объявление, оно открывается и можно увидеть более подробную информацию про автомобиль, также визу находится кнопка «Позвонить!», при нажатии на которую можно позвонить человеку, который оставил объявление о продаже. Также вверху находится корзина, чтобы пользователь, оставивший объявление мог его удалить, если оно больше не актуально. Страницу объявления можно увидеть на рис.4.

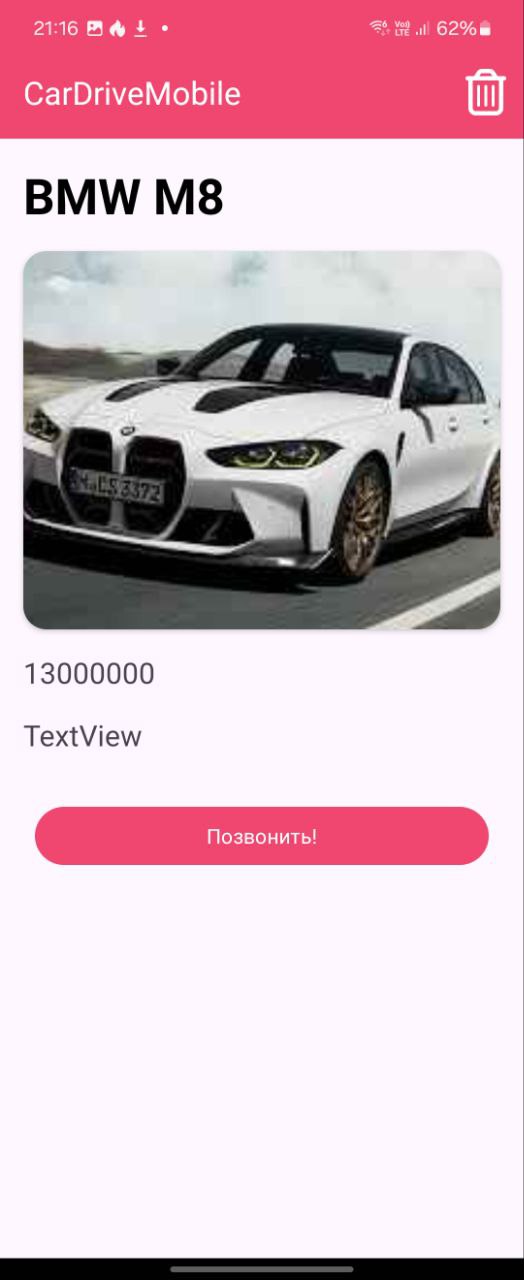


Рис.4 – Страница объявления.

На странице создания объявления (рис.5) нужно указать название, цену и двигатель, также можно добавить фото, после чего можно нажать кнопку «Создать».

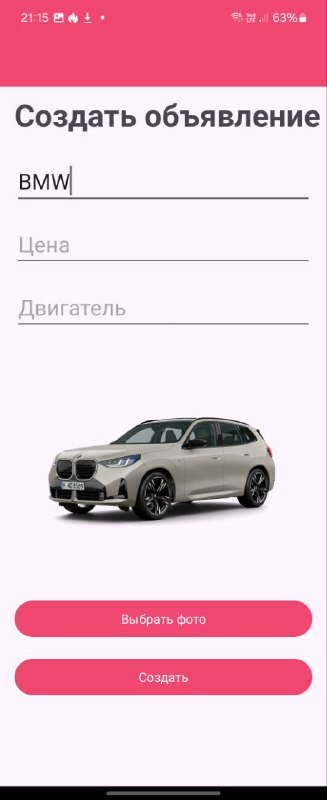


Рис.5 – Создание объявления.

На странице личного кабинета (рис.6) можно изменить персональные данные, такие как номер телефона и пароль. Также вверху есть кнопка выхода из аккаунта.

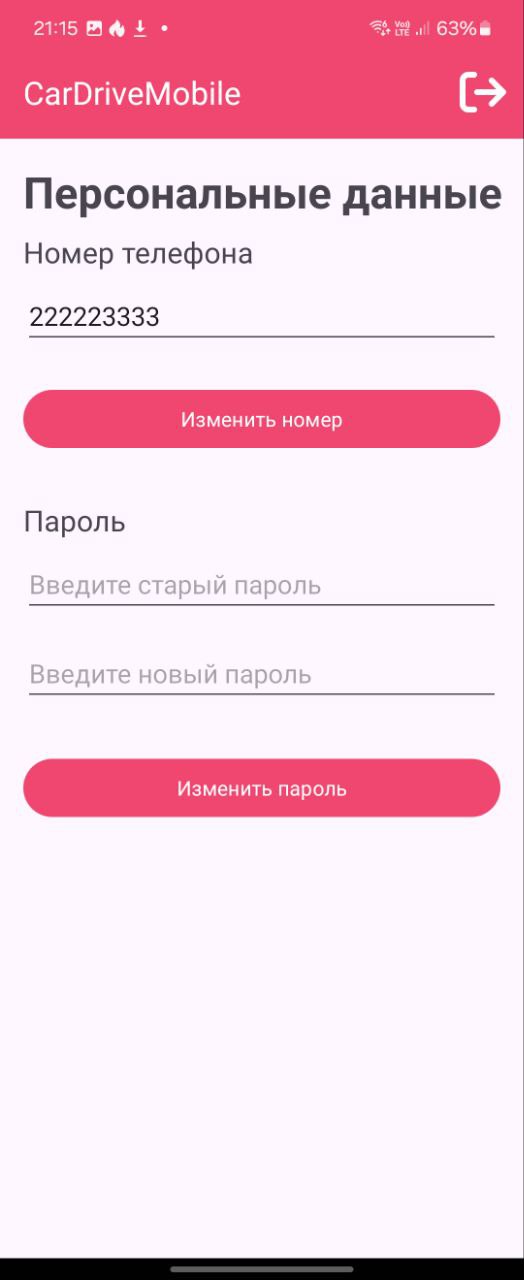


Рис.6 – Личный кабинет.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы была проведена разработка мобильного приложения для магазина автомобилей. Основное внимание уделялось созданию удобной и эффективной платформы, которая позволила бы пользователям легко находить интересующую их информацию о транспортных средствах.

Важнейшей частью проекта стала реализация клиент-серверной архитектуры, которая обеспечивает высокую степень надежности. Для разработки клиентской части использовался язык программирования Java и среда разработки Android Studio, что позволило создать интуитивно понятный и удобный интерфейс для пользователей. Серверная часть была выполнена на языке PHP с использованием СУБД MySQL, что обеспечило быструю и безопасную обработку данных.

Разработанное приложение демонстрирует высокие функциональные возможности, включая авторизацию пользователей, просмотр каталога автомобилей и детальную информацию о каждой машине. Это позволит потенциальным покупателям быстро найти нужный автомобиль, что способствует росту продаж и повышению уровня обслуживания клиентов.

Таким образом, данное исследование показало, что разработка мобильных приложений – это перспективное направление, способствующее развитию бизнеса и улучшению взаимодействия с клиентами. Полученные знания и навыки могут быть применены в будущем для создания аналогичных продуктов, направленных на удовлетворение потребностей современных пользователей.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Бобровский, Н.В. Основы разработки мобильных приложений [Текст] / Н.В. Бобровский. – СПб.: Питер, 2018. – 640 с.
2. Гриненко, А.А. Программирование на Java для мобильных устройств [Текст] / А.А. Гриненко. – М.: Директ-Медиа, 2017. – 384 с.
3. Петров, А.С. Разработка мобильных приложений на Android [Текст] / А.С. Петров. – М.: Альфа-М, 2018. – 352 с.
4. Петренко, Ю.П. Разработка серверных приложений на PHP [Текст] / Ю.П. Петренко. – М.: Инфра-М, 2019. – 480 с.