

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 продажи и покупки автомобилей».

Пояснительная записка

Листов: 63

Руководитель

_____ / М.А. Горбунова

« ____ » _____ 2025 год

Исполнитель

_____ / И.А. Пушкин

« ____ » _____ 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)

Пушкиным Ильей Александровичем

(фамилия, имя, отчество студента — полностью)

студенту группы П50-3-22 специальности 09.02.07 «Информационные системы и программирование» по МДК 01.01 «Разработка программных модулей»

1. Исходные данные к проекту (работе):
 - 1.1. Тема: «Разработка мобильного приложения для продажи и покупки автомобилей».
 - 1.2. Состав курсового проекта:
 - 1.2.1. Задание КП
 - 1.2.2. Пояснительная записка
 - 1.2.3. Программа (исходные данные) на электронном носителе
 - 1.2.4. Презентация и инсталляционный пакет программы на электронном носителе
 - 1.3. Содержание пояснительной записки:

ВВЕДЕНИЕ

 1. ОБЩАЯ ЧАСТЬ
 - 1.1. Цель разработки
 - 1.2. Средства разработки
 2. СПЕЦИАЛЬНАЯ ЧАСТЬ
 - 2.1. Постановка задачи
 - 2.1.1. Входные данные
 - 2.1.2. Выходные данные
 - 2.1.3. Подробные требования к проекту
 - 2.2. Внешняя спецификация
 - 2.2.1. Описание задачи
 - 2.2.2. Входные и выходные данные
 - 2.2.3. Методы
 - 2.2.4. Тесты
 - 2.2.5. Контроль целостности данных
 - 2.3. Проектирование
 - 2.3.1. Схема архитектуры приложения
 - 2.3.2. Логическая схема данных
 - 2.3.3. Физическая схема данных
 - 2.3.4. Структурная схема
 - 2.3.5. Функциональная схема
 - 2.3.6. Диаграмма классов
 - 2.3.7. Схема тестирования

2.3.8. Схема пользовательского интерфейса

2.4.Результат работы программы

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1.Инструментальные средства

3.2.Отладка программы

3.3.Защитное программирование

3.4.Характеристики программы

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

ПРИЛОЖЕНИЕ А. Техническое задание

ПРИЛОЖЕНИЕ Б. Текст программы

ПРИЛОЖЕНИЕ В. Сценарий и результаты тестовых испытаний

ПРИЛОЖЕНИЕ Г. Руководство пользователя

2. 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	
1.	Введение	-	
2.	Цель разработки	-	
3.	Средства разработки	-	
Б	Анализ задачи и её постановка	15	
1.	Определение требований к программе	-	
2.	Спецификация программы (описание задачи, описание входных и выходных данных, методы)	-	
3.	Тесты, контроль целостности данных	-	
В	Проектирование и реализация	55	
1.	Схемы проекта (схема архитектуры, функциональная и структурная схемы, диаграмма классов, схема тестирования, схема пользовательского интерфейса)	-	
2.	Реализация в инструментальной среде	-	
Г	Технологическая часть проекта	5	
1.	Инструментальные средства разработки	-	
2.	Отладка программа	-	
3.	Защитное программирование	-	
4.	Характеристика программы	-	
Д	Программная документация	10	
1.	Приложение А. Техническое задание	-	
2.	Приложение Б. Текст программы	-	
3.	Приложение В. Сценарий и результаты тестовых испытаний	-	
	Приложение Г. Руководство пользователя		

	Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения
Е	Экспериментальная часть проекта	10	
1.	Программа на машинном носителе. Информация на носителе разбита на разделы: эксплуатационный пакет, тексты программы, документация.	-	

Руководитель курсового проекта (работы) Горбунова Мария Александровна, преподаватель

«22» февраля 2025 года _____ / М.А. Горбунова/

Дата выдачи курсового задания «04» февраля 2025 года

Срок сдачи законченного проекта (работы) «22» апреля 2025 года

Задание принял к исполнению

«04» февраля 2025 года _____ / И.А. Пушкин /

СОДЕРЖАНИЕ

Введение	4
1. Общая часть	6
1.1. Цель разработки	6
1.2. Средства разработки	6
2. Средства разработки.....	8
2.1. Постановка задачи	8
2.1.1. Входные данные	8
2.1.2. Выходные данные предметной области.....	8
2.1.3. Подробные требования к проекту	9
2.2. Внешняя спецификация.....	10
2.2.1. Описание задачи.....	10
2.2.2. Входные и выходные данные	21
2.2.3. Методы.....	25
2.2.4. Тесты	25
2.2.5. Контроль целостности данных	26
2.3. Проектирование	27
2.3.1. Схема архитектуры приложения	27
2.3.2. Логическая схема данных	27
2.3.3. Физическая схема данных.....	27
2.3.4. Структурная схема	27
2.3.5. Функциональная схема	28
2.3.6. Диаграмма классов	29
2.3.7. Схема тестирования	32
2.3.8. Схема пользовательского интерфейса	33

2.4. Результат работы программы	34
3. Технологическая часть	41
3.1. Инструментальные средства	41
3.2. Отладка программы	41
3.3. Защитное программирование	42
3.4. Характеристики программы	44
Заключение	58
Список использованных материалов	60

ВВЕДЕНИЕ

В условиях активного развития информационных технологий и роста популярности мобильных сервисов пользователи всё чаще отдают предпочтение мобильным приложениям для решения повседневных задач, включая покупку, продажу и управление личной информацией. Создание удобных и многофункциональных мобильных платформ стало важной задачей для улучшения пользовательского опыта и расширения возможностей цифровой коммерции. Мобильные приложения объединяют продавцов, покупателей и администраторов в едином пространстве, обеспечивая быстрое и интуитивно понятное взаимодействие. Разработка подобного приложения позволяет значительно повысить доступность сервисов и упростить бизнес-процессы.

Мобильное приложение «Drivee» представляет собой современную платформу для Android-устройств, предназначенную для продажи, покупки автомобилей, а также управления личными объявлениями и записями в бортжурнале. Система объединяет частных продавцов, покупателей и администраторов, создавая удобную среду для поиска транспортных средств, подачи объявлений и проведения модерации. В отличие от традиционных интернет-площадок, приложение «Drivee» оптимизировано для быстрого взаимодействия, что делает процесс выбора автомобиля простым, быстрым и удобным для пользователя.

Приложение ориентировано на пользователей, активно использующих мобильные устройства для повседневных задач. Частные лица могут легко находить интересующие их автомобили, применять расширенные фильтры поиска, добавлять объявления в избранное, а продавцы — публиковать и редактировать свои предложения, управлять личным профилем и следить за состоянием своих объявлений. Администраторы получают возможность модерировать контент, управлять марками и моделями автомобилей, а также контролировать записи бортжурнала, что делает платформу эффективной и безопасной для всех участников.

В рамках курсового проекта будет разработано мобильное приложение «Drivee» в соответствии с техническим заданием (Приложение А. Техническое задание). Целью разработки является создание комплексной системы, включающей основные модули: регистрация и авторизация пользователей, управление личным профилем, поиск и фильтрацию объявлений, управление избранным, публикацию и редактирование объявлений, работу с боржурналом и администрирование контента. Эти компоненты обеспечат полный цикл взаимодействия пользователя с платформой.

Приложение «Drivee» значительно упростит процесс поиска, продажи и покупки автомобилей, предоставив пользователям удобный, интуитивный интерфейс и надежные инструменты для взаимодействия. Платформа оптимизирует рабочие процессы продавцов, облегчает покупку автомобилей для частных лиц и обеспечивает эффективную модерацию контента администраторами, повышая общее качество предоставляемых услуг и удовлетворенность пользователей.

1. ОБЩАЯ ЧАСТЬ

1.1. Цель разработки

Создание функционального мобильного приложения «Drivee», которое обеспечит удобное взаимодействие между покупателями и продавцами в сфере торговли автомобилями. Проект направлен на повышение доступности автомобильной продукции, улучшение пользовательского опыта и оптимизацию процессов купли-продажи для всех участников платформы.

1.2. Средства разработки

В качестве программных средств разработки использовалось ПО, представленное в Таблице 1 (см. Таблица 1 – Программные средства разработки ПО)

Таблица 1 – Программные средства разработки ПО

№	Тип средства	Наименование средства	Назначение
1	2	3	4
1	Операционная система	MacOS 15.3.2 (24D81)	Организация взаимодействия программ и пользователя
2	Инструментальные средства разработки программного решения	Android Studio	Разработка мобильного приложения
3	Облачная база данных	Google Firestore Database	Хранение и управление данными мобильного приложения в облаке
5	Сервис электронной почты	Firebase	Отправка писем для восстановления пароля и регистрации пользователей, а также для получения чека заказа
6	Текстовый редактор	Microsoft® Word для Mac	Написание документации
7	Графический построитель	Pixso	Создание эскиза визуальной части мобильного приложения

В качестве средств вычислительной техники для проектирования, разработки и тестирования программного комплекса использовались ноутбук MacBook air 15 M3, а также мобильное устройство Xiaomi 4 note. Характеристики технических средств представлены в Таблице 2 (см. Таблица 2 – Характеристики вычислительной техники).

Таблица 2 - Характеристики вычислительной техники

№	Характеристика	Значение
1	2	3
MacBook Air 15 M3		
1	Разрешение экрана	2880 × 1864
2	Видеокарта	Apple M3
3	Размер экрана	15.3 " дюйма
4	Оперативная память	16 ГБ
5	Графический процессор	Apple M3
Xiaomi 4 note		
1	Дисплей	7.5" (2430x1080), Full HD, Super AMOLED
2	Встроенная память	128 ГБ
3	Оперативная память	4 ГБ
4	Процессор	Snapdragon 8g
5	Количество ядер процессора	6

2. СРЕДСТВА РАЗРАБОТКИ

2.1. Постановка задачи

Целью данного проекта является разработка мобильного приложения «Drivee» для платформы Android, предназначенного для автоматизации процессов публикации, поиска и продажи автомобилей. Приложение должно обеспечивать удобное взаимодействие между покупателями, продавцами и администраторами, предоставляя функционал для частной и коммерческой продажи автомобилей. Оно ориентировано на частных пользователей, автосалоны и сервисные компании, объединяя их в единой экосистеме.

2.1.1. Входные данные

Мобильное приложение «Drivee» содержит следующие входные данные: информация о пользователях (имя, дата рождения, номер телефона, электронная почта, пароль, фото профиля), автомобилях (марка, модель, год выпуска, цвет, тип кузова, объем двигателя, мощность, тип привода, тип топлива, трансмиссия, пробег, цена, описание, фотографии автомобиля), объявлениях (статус модерации, дата публикации), данных боржурнала (марка и модель автомобиля, заголовок записи, текст записи, пробег) и параметрах фильтрации поиска (бренд, модель, цена, пробег, город, год выпуска и прочее).

2.1.2. Выходные данные предметной области

Мобильное приложение «Drivee» содержит следующие выходные данные: о пользователях (информация о профиле: имя, дата рождения, номер телефона, электронная почта, пароль, фотография профиля), об автомобилях (марка, модель, год выпуска, цвет, тип кузова, объем двигателя, мощность, пробег, тип привода, тип топлива, трансмиссия, цена, описание, фотографии автомобилей), об объявлениях (статус модерации: на проверке, подтверждено; дата публикации, идентификатор объявления), о записях в боржурнале (марка и модель автомобиля, заголовок записи, текст записи, пробег автомобиля на момент записи), о фильтрации объявлений (список

отфильтрованных автомобилей по марке, модели, цене, пробегу, городу, цвету, типу кузова, приводу, типу топлива, трансмиссии).

Также приложение формирует информацию о действиях пользователя, таких как добавление/удаление машины в избранное, удаление или изменение объявлений, изменения профиля пользователя.

2.1.3. Подробные требования к проекту

Разработать мобильное приложение «Drivee» для платформы Android с использованием облачной СУБД Firebase (Firestore) для хранения и обработки данных. Приложение должно обеспечивать пользователям возможность просматривать объявления о продаже автомобилей, использовать систему фильтрации по различным параметрам и управлять личными данными и объявлениями.

Для доступа ко всем функциям приложения требуется авторизация. Регистрация новых пользователей производится с указанием имени, даты рождения, номера телефона, email и пароля (пароль должен содержать не менее 8 символов, включая строчные, заглавные буквы и цифры). В случае утери пароля предусмотрена возможность его восстановления через email с использованием SMTP-сервера.

Интерфейс приложения должен быть интуитивно понятным, с удобной навигацией через нижнюю панель (главная, избранное, создать объявление, боржурнал, профиль), обеспечивая пользователю быстрый доступ к основным функциям. Приложение «Drivee» должно поддерживать следующие функциональные возможности:

- Возможность авторизации и регистрации пользователей.

- Возможность восстановления пароля через SMTP.

- Возможность изменения данных профиля (фото профиля, имя, телефон, email).

- Возможность создания объявления о продаже автомобиля.

- Возможность загрузки фотографий автомобиля (до 5 штук).

Возможность выбора параметров автомобиля (марка, модель, год, объем двигателя, цвет, мощность, пробег, тип привода, топливо, трансмиссия, цена, описание).

Возможность редактирования созданного объявления.

Возможность удаления своего объявления.

Возможность просмотра всех объявлений.

Возможность применения фильтров поиска по марке, модели, году, цене, объему двигателя, мощности, количеству владельцев, цвету, кузову, городу, приводу, типу топлива, типу трансмиссии.

Возможность добавления автомобиля в избранное.

Возможность удаления автомобиля из избранного.

Возможность создания записи в бортижурнале по автомобилю (марка, модель, заголовок записи, текст записи, пробег).

Возможность редактирования и удаления записей бортижурнала.

Возможность просмотра записей бортижурнала в ленте.

Возможность выхода из аккаунта пользователя.

Возможность работы администратора для модерирования объявлений и записей бортижурнала.

Возможность подтверждения или удаления объявлений администратором.

Возможность управления списком марок и моделей автомобилей администратором (добавление новых марок и моделей).

2.2. Внешняя спецификация

2.2.1. Описание задачи

Разработать мобильное приложение для цифровизации процесса покупки, продажи и администрирования объявлений в сфере автомобильной торговли. Решение включает использование облачной базы данных, управляемой СУБД Firebase (Firestore), и разработку мобильного приложения на платформе Android через Android Studio.

Приложение должно обеспечивать не только отображение данных, но и предоставлять пользователям возможности регистрации, авторизации, восстановления пароля, создания, редактирования и удаления объявлений, управления личными данными, а также работы с избранными объявлениями и бортжурналом.

На этапе анализа были определены основные пользовательские роли: неавторизованный пользователь, покупатель и администратор, а также описан их функционал в системе: регистрация и вход в систему, сброс пароля, просмотр объявлений, фильтрация по параметрам, добавление автомобилей в избранное, создание и управление объявлениями, публикация записей в бортжурнале, модерирование контента (объявлений и записей) администраторами, управление марками и моделями автомобилей.

На Рисунке 1 показана диаграмма прецедентов.

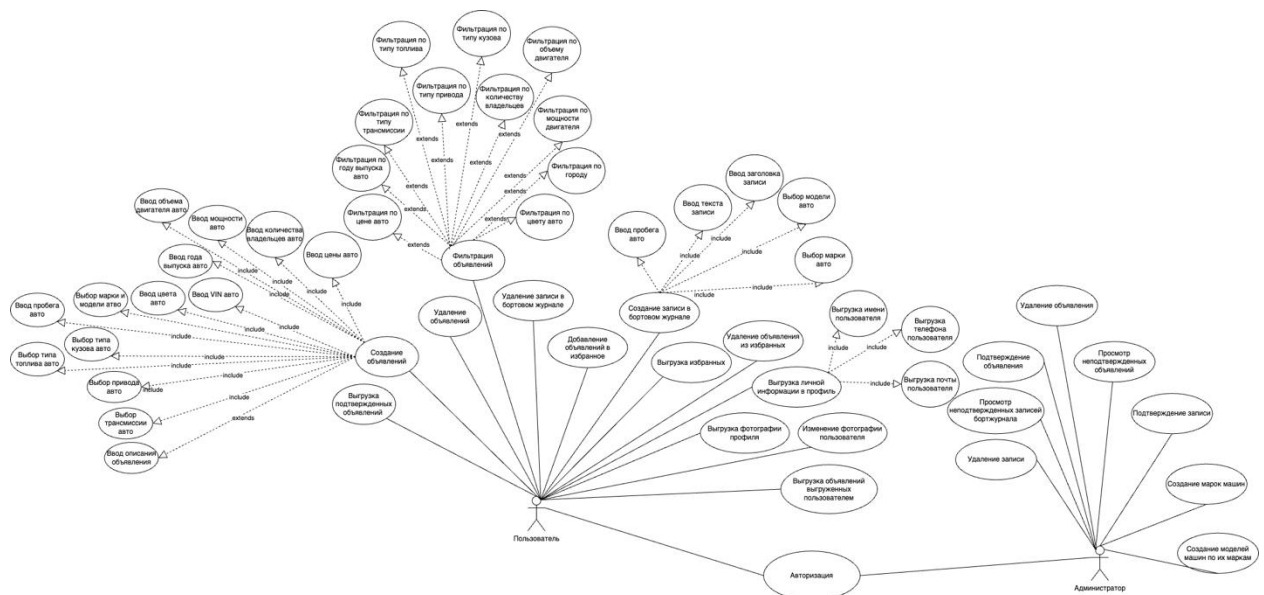


Рисунок 1 - Диаграмма прецедентов

На основе созданной диаграммы прецедентов можно рассмотреть возможности каждого пользователя:

Пользователь с ролью «Неавторизованный пользователь» имеет следующие возможности:

Регистрация в качестве пользователя с заполнением следующих полей: почта, имя, дата рождения, номер телефона, пароль, подтверждение пароля.

Авторизация в системе с заполнением полей: почта и пароль.

Восстановление пароля через отправку ссылки для сброса пароля на почту через SMTP.

Пользователь с ролью «Обычный пользователь» имеет следующие возможности:

Авторизация через заполнение полей: почта и пароль.

Восстановление пароля через отправку ссылки для сброса пароля на почту через SMTP.

Просмотр списка всех доступных объявлений о продаже автомобилей в виде карточек (фото, марка, модель, цена, пробег).

Применение фильтрации объявлений по марке, модели, цене, пробегу и дополнительным параметрам.

Сортировка объявлений по цене (по возрастанию/убыванию) и новизне.

Добавление и удаление автомобилей в избранное.

Просмотр избранных автомобилей.

Создание новых объявлений о продаже автомобиля с заполнением данных: VIN, марка, модель, год выпуска, цвет, объем двигателя, мощность, пробег, количество владельцев, цена, описание, город, тип кузова, трансмиссия, тип топлива, тип привода, а также загрузка до 5 фотографий.

Просмотр своих объявлений в профиле.

Удаление своих объявлений.

Просмотр профиля пользователя (имя, телефон, email).

Изменение фотографии профиля.

Выход из аккаунта.

Работа с боржурналом: создание записей о техническом состоянии автомобиля с заполнением полей: марка, модель, заголовок, текст записи, пробег.

Просмотр записей боржурнала.

Пользователь с ролью «Администратор» имеет следующие возможности:

Авторизация через заполнение полей: почта и пароль.

Восстановление пароля через отправку ссылки на почту через SMTP.

Просмотр всех неподтвержденных объявлений.

Принятие или удаление объявлений.

Просмотр всех записей боржурнала.

Принятие или удаление записей боржурнала.

Управление списком марок и моделей автомобилей: добавление новых марок и моделей.

Удаление записей или объявлений по необходимости.

Выход из аккаунта администратора.

Перед созданием мобильного приложения «Drivee» был выполнен анализ предметной области для определения ключевых процессов.

В результате были выделены процессы, подлежащие автоматизации.

На Рисунках 2–8 представлена диаграмма бизнес-процессов в нотации IDEF0, описывающая состояние перед внедрением приложения покупки и продаж автомобилей «Drivee».

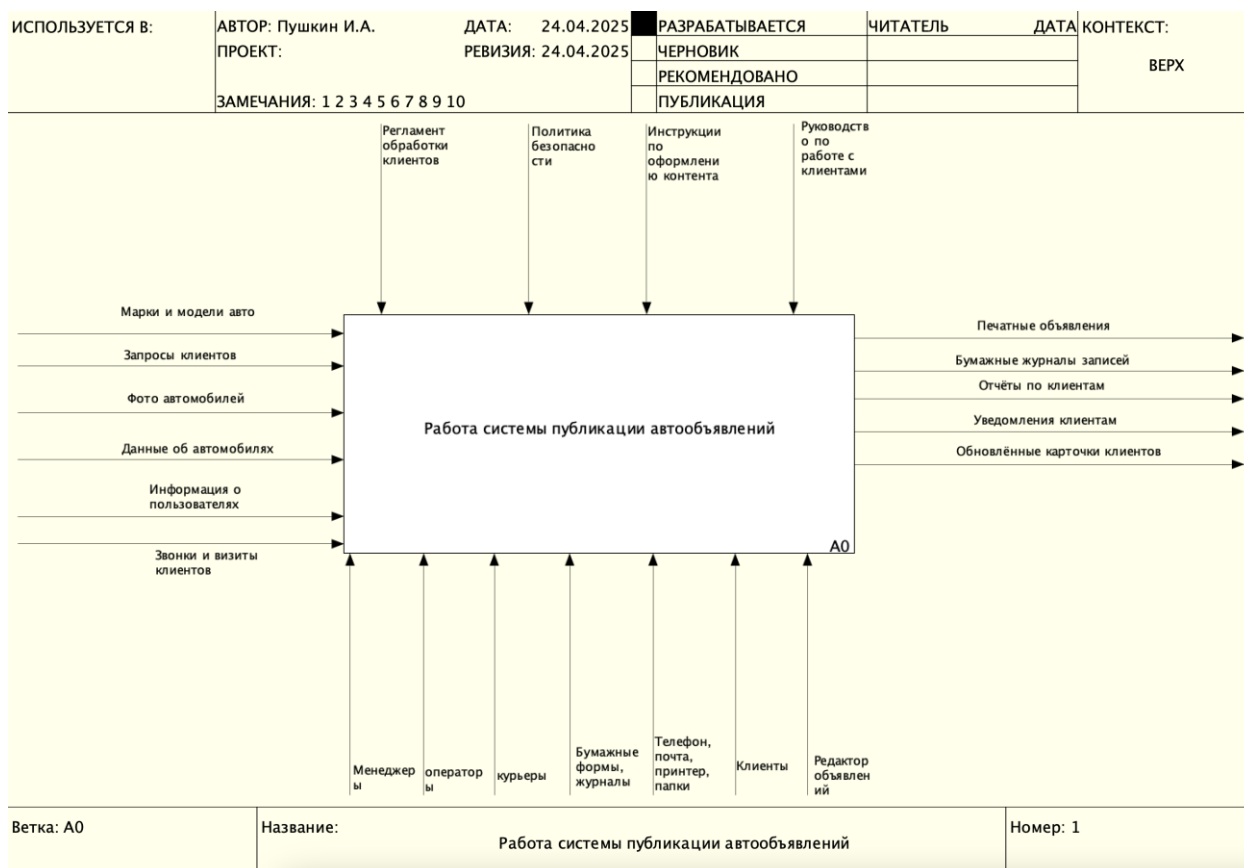


Рисунок 2 - Подпроцесс работы приложения ДО автоматизации

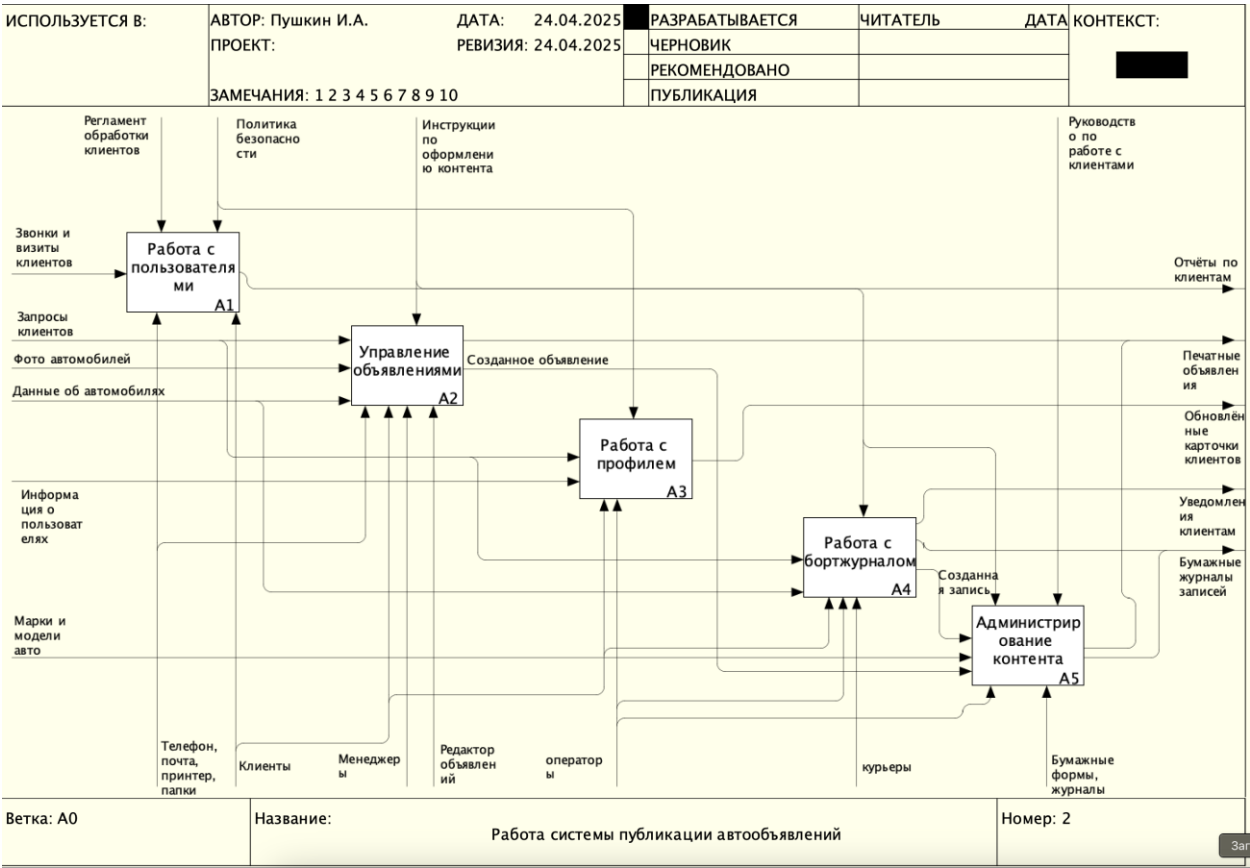


Рисунок 3 - Декомпозиция основного уровня

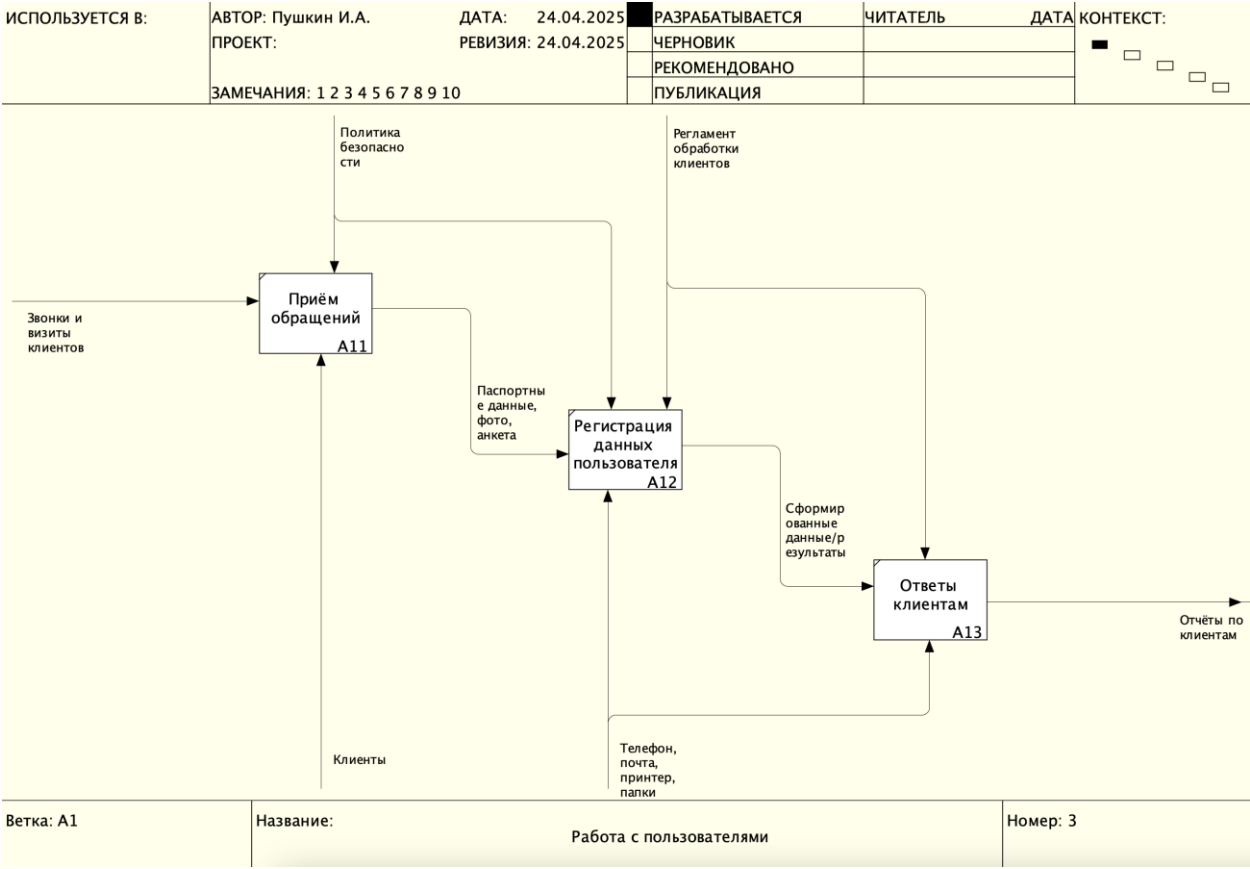


Рисунок 4 – Декомпозиция подпроцесса работы с пользователями

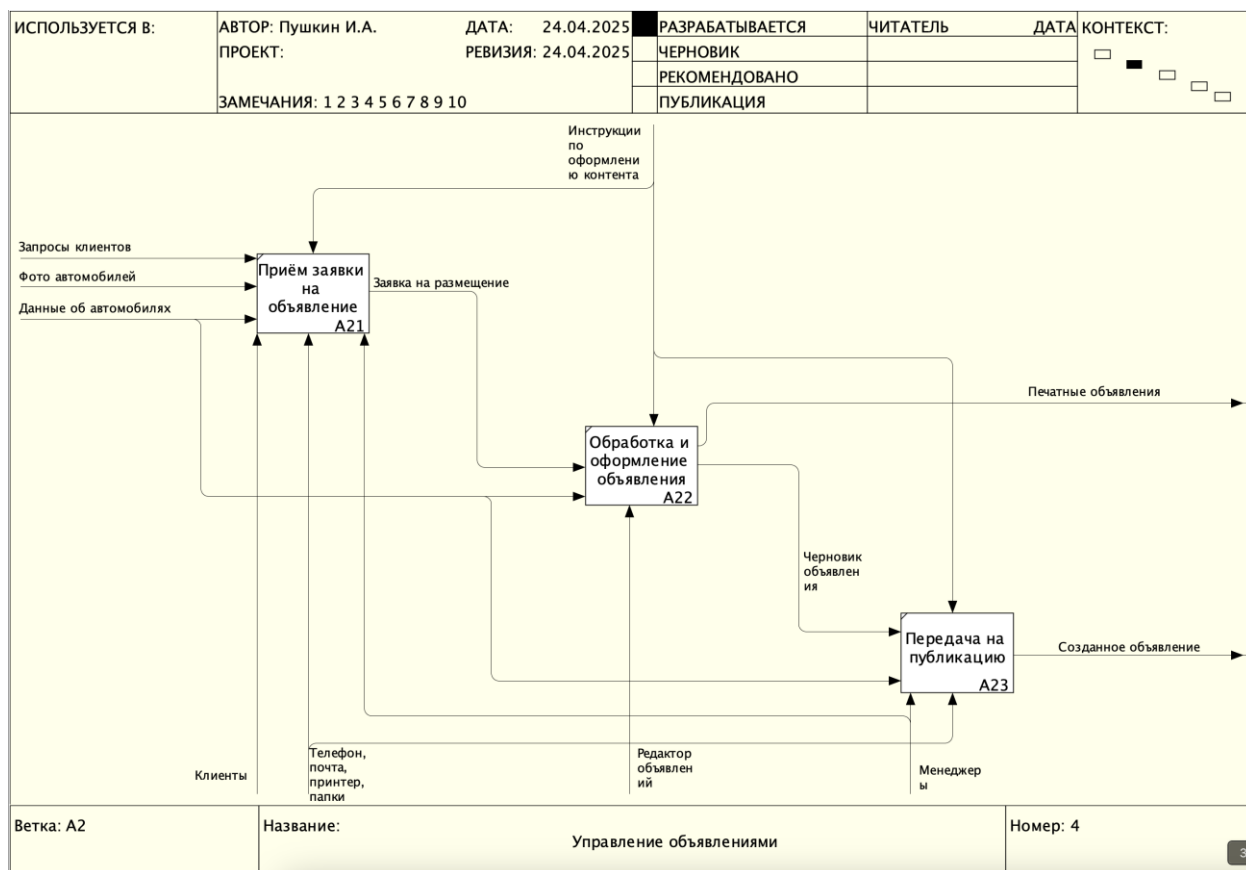


Рисунок 5 - Декомпозиция подпроцесса управления объявления

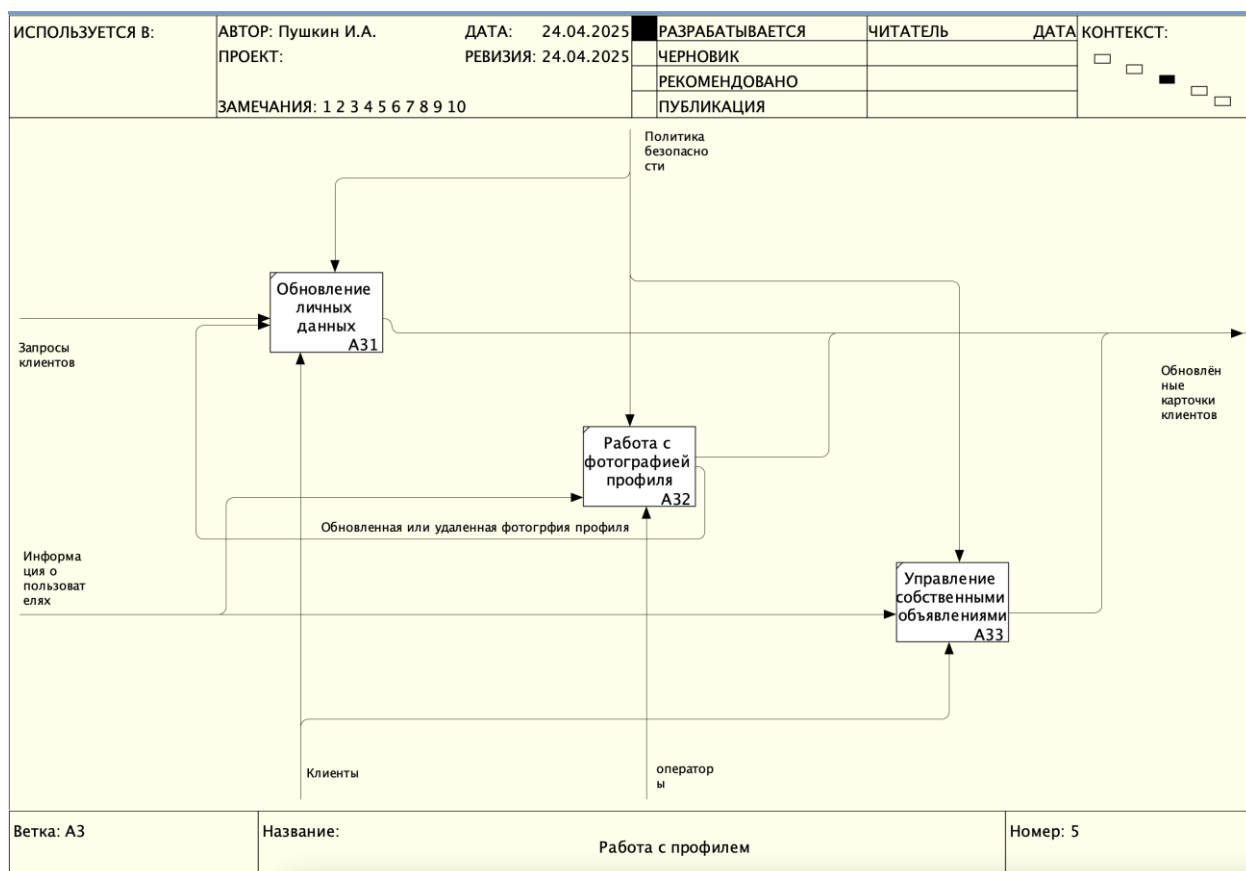


Рисунок 6 - Декомпозиция подпроцесса работы с профилем

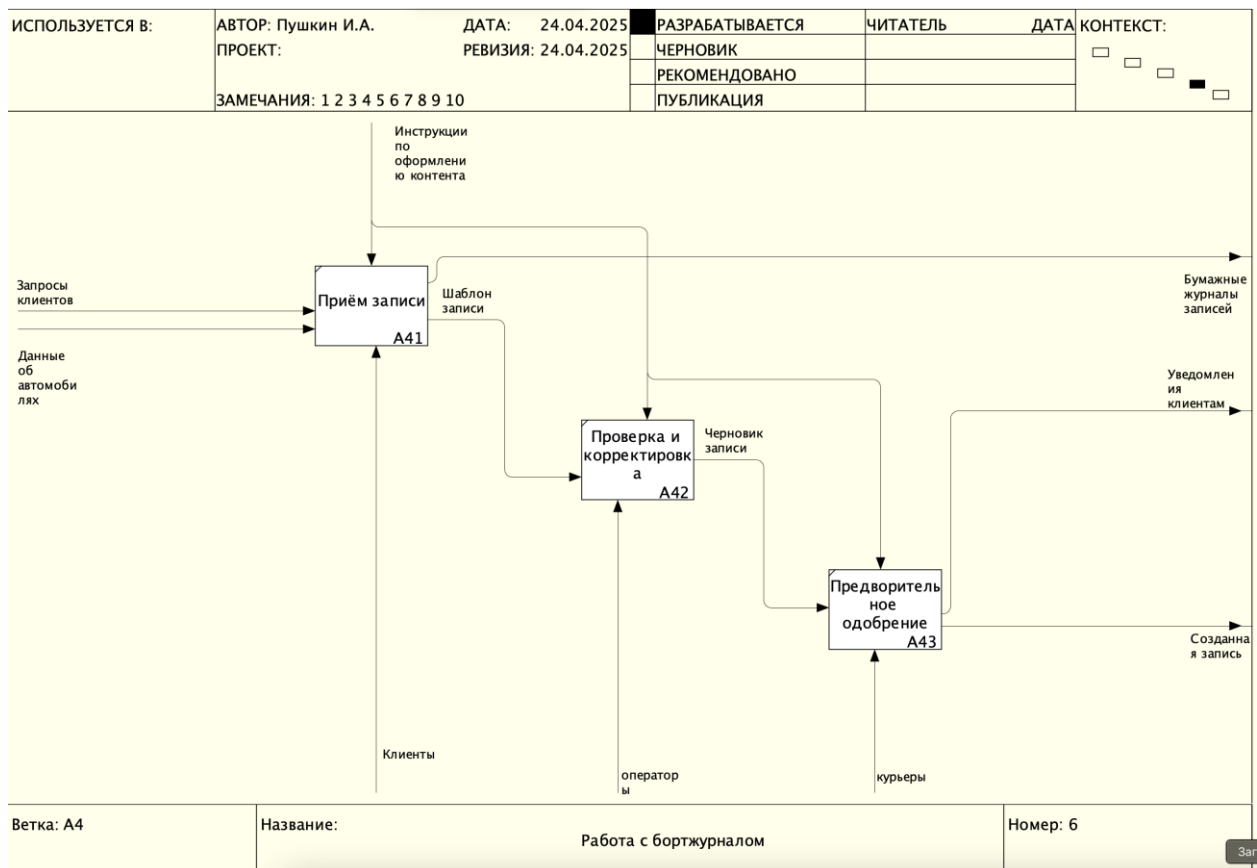


Рисунок 7 - Декомпозиция подпроцесса работы с бортижурналом

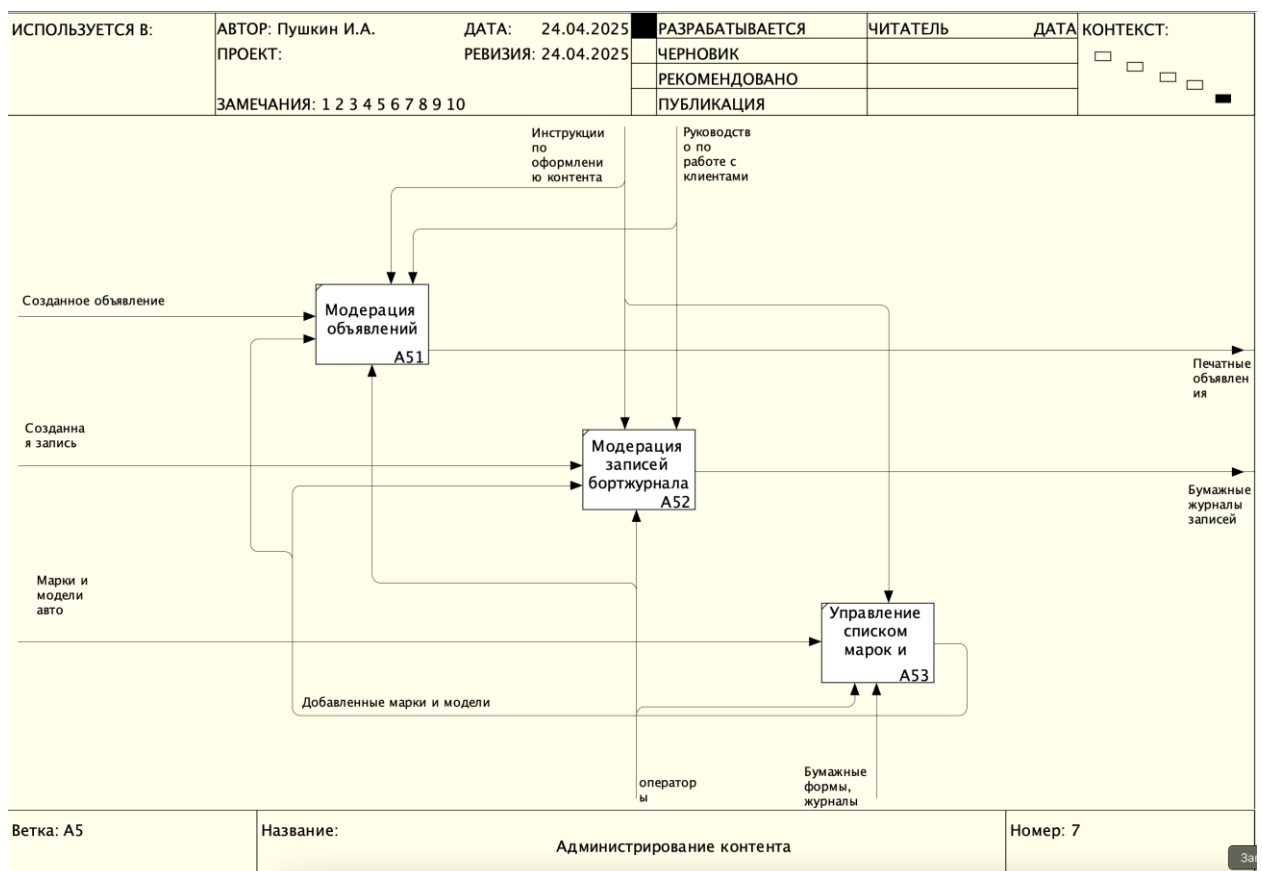


Рисунок 8 - Декомпозиция подпроцесса администрирования контента

На Рисунках 9–15 представлена диаграмма бизнес-процессов в нотации IDEF0, описывающая состояние после внедрения приложения покупки и продаж автомобилей «Drivee».

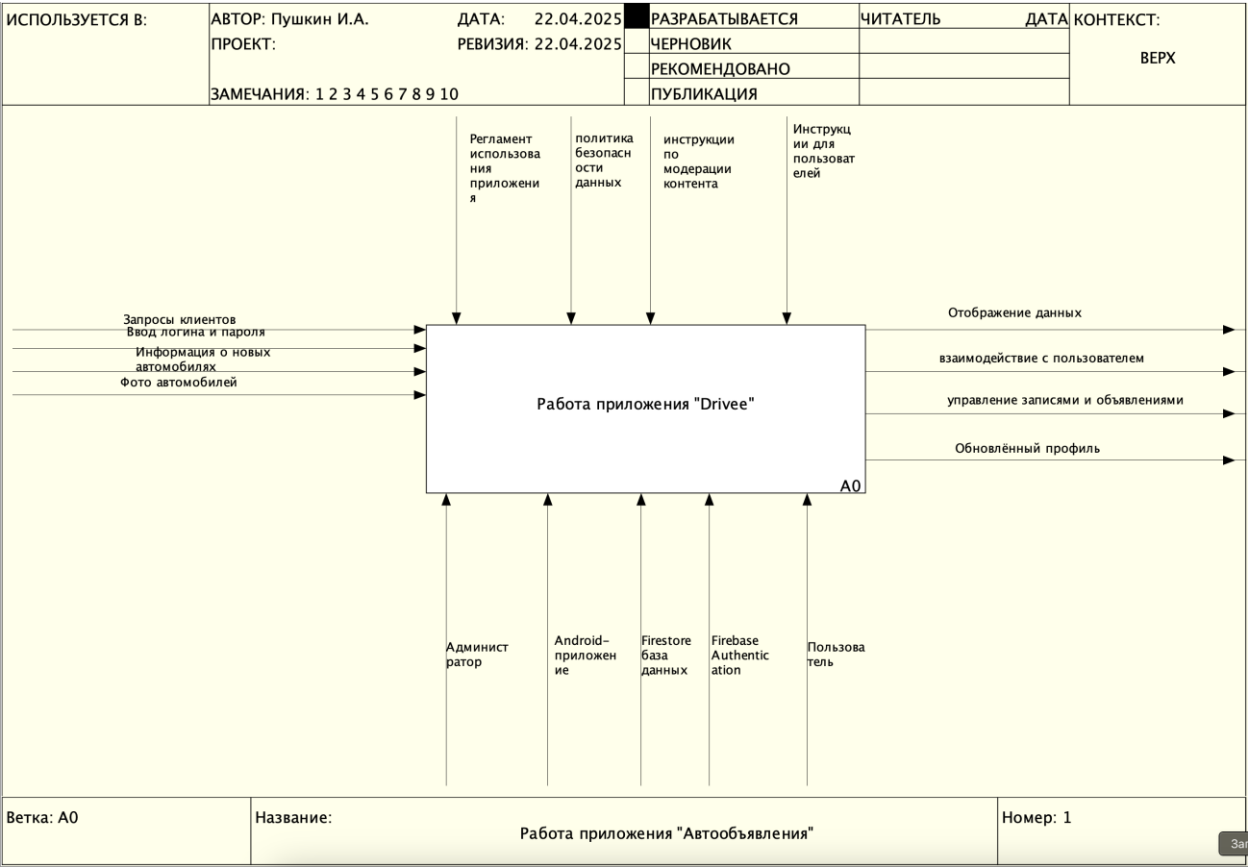


Рисунок 9 – Подпроцесс работы приложения ПОСЛЕ автоматизации

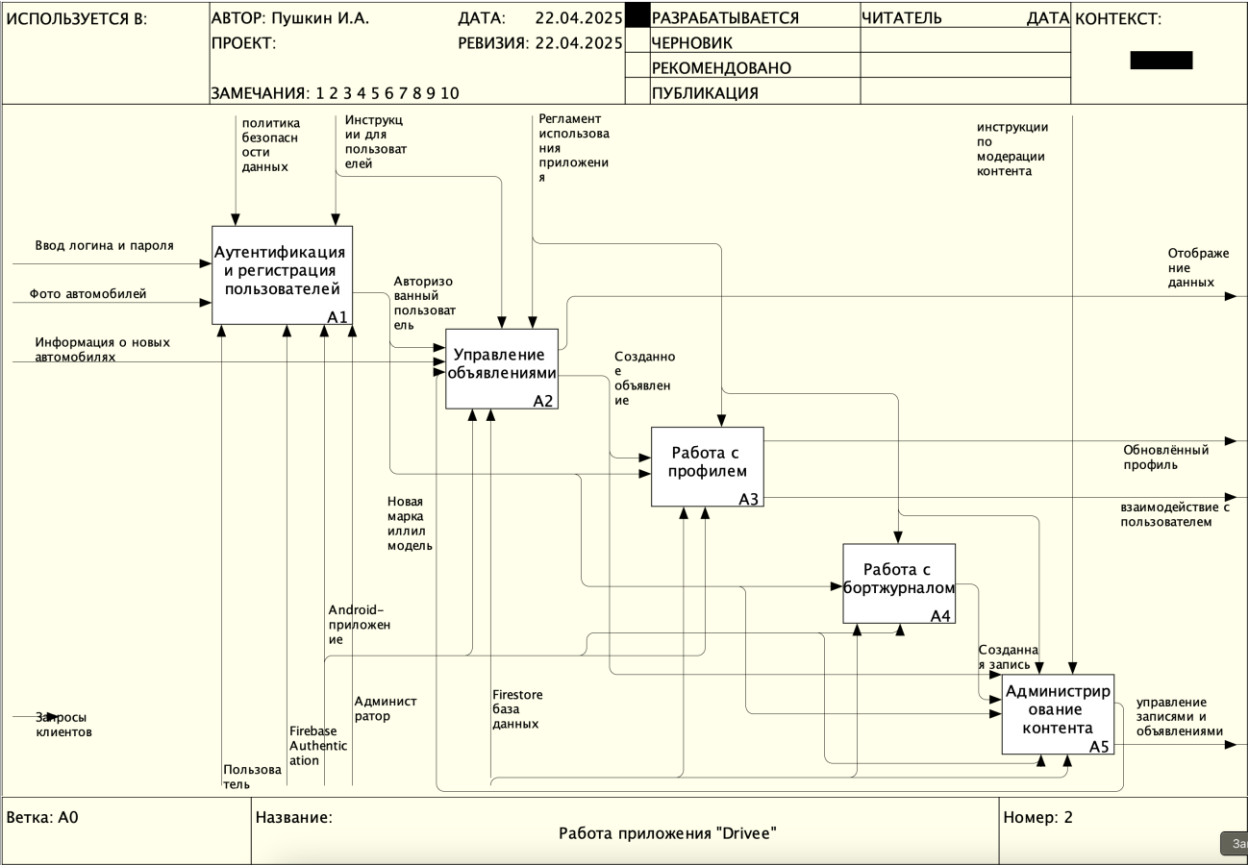


Рисунок 10 – Декомпозиция основного уровня

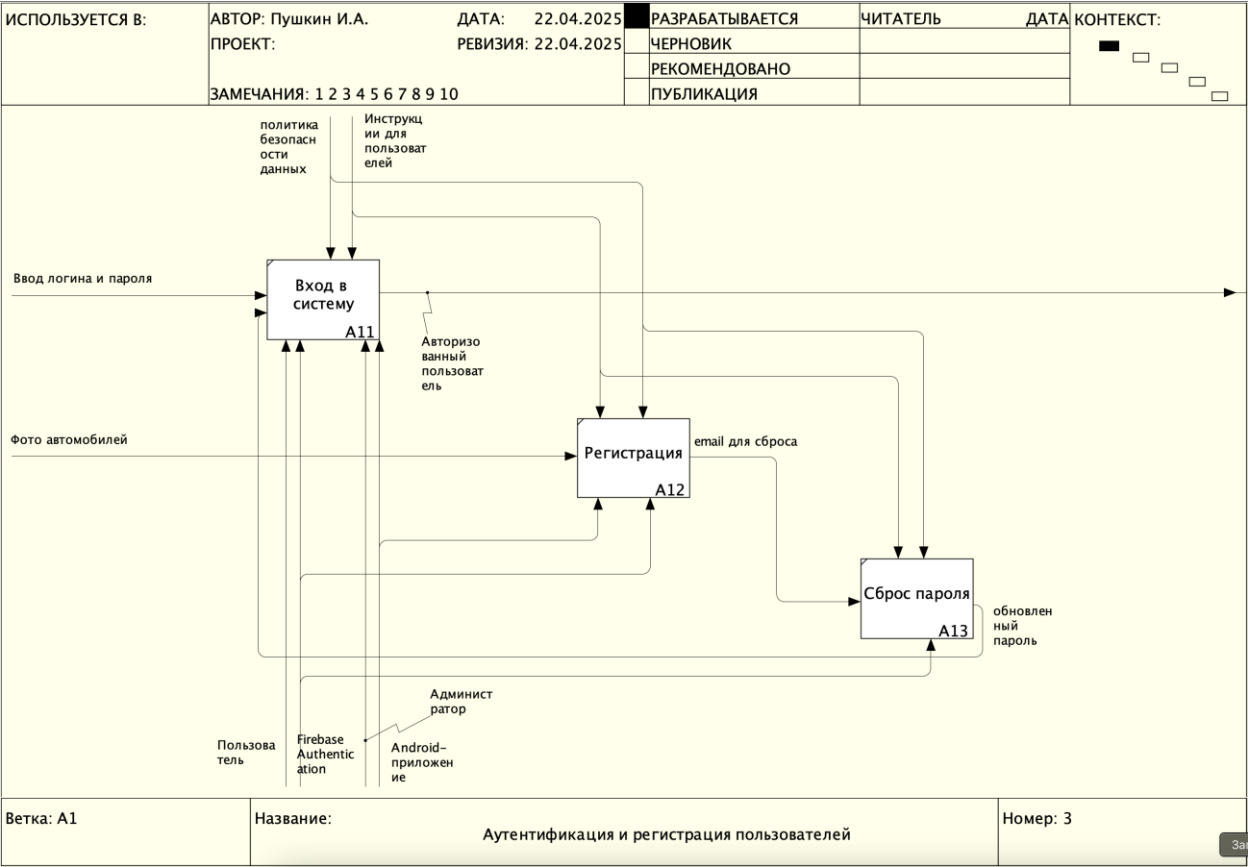


Рисунок 11 – Декомпозиция аутентификации и регистрации пользователей

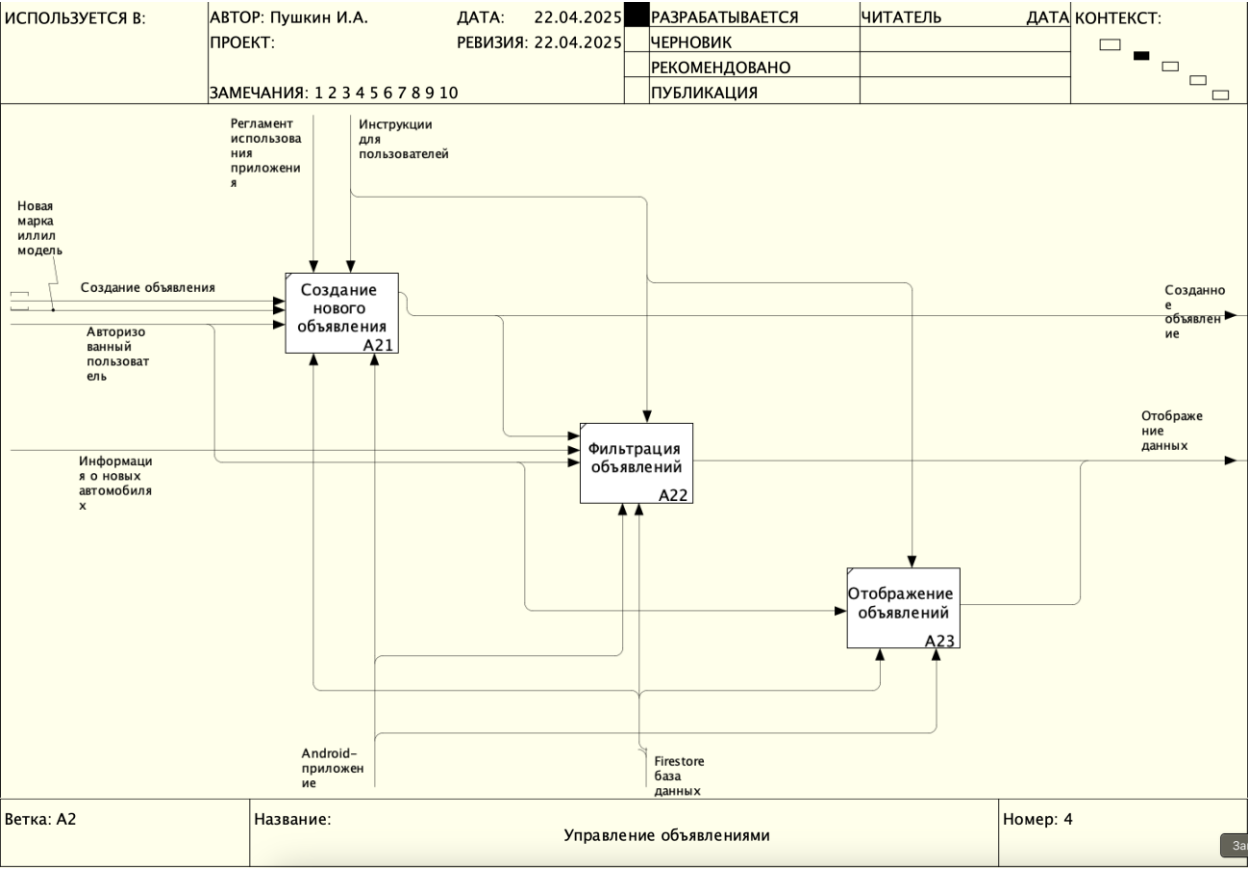


Рисунок 12 – Декомпозиция управления объявлениями

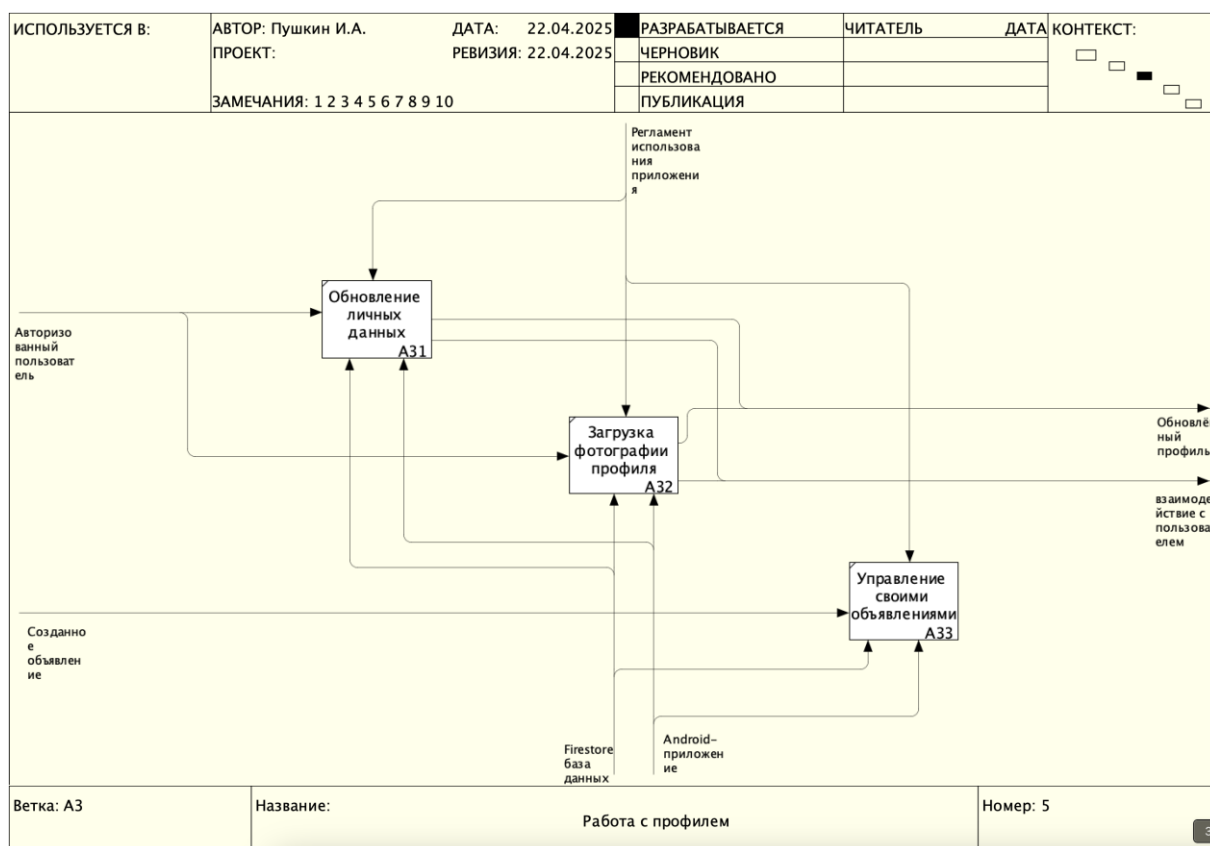


Рисунок 13 – Декомпозиция работы с профилем

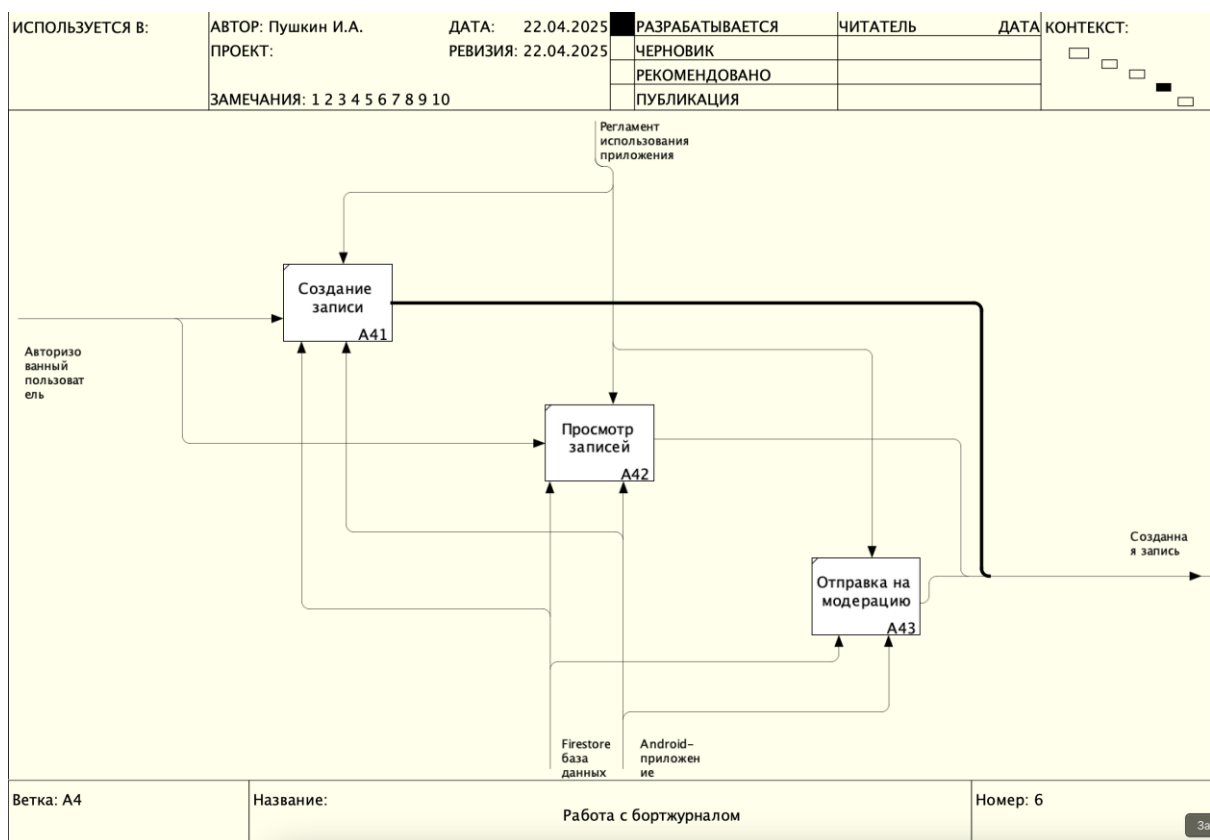


Рисунок 14 – Декомпозиция работы с боржурналом

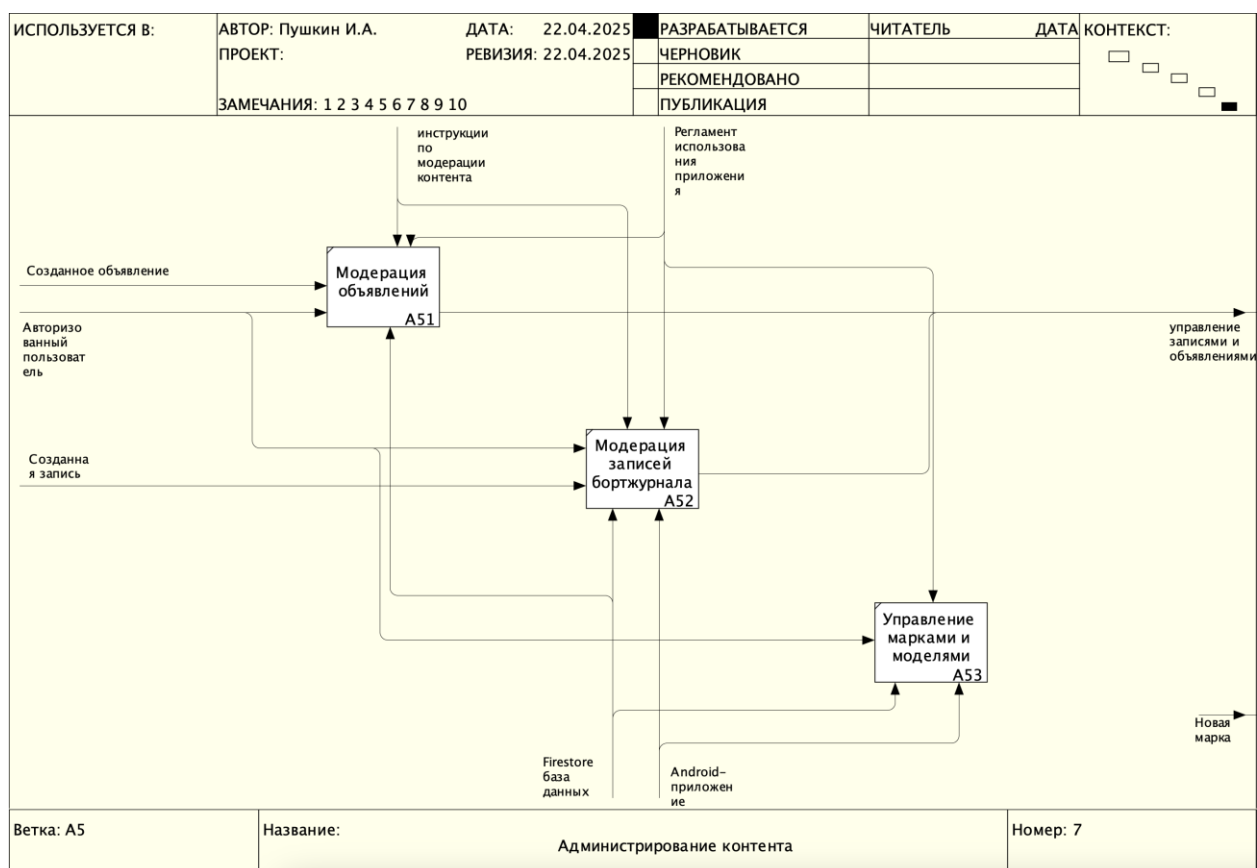


Рисунок 15 – Декомпозиция администрирования контента

2.2.2. Входные и выходные данные

В Таблице 3 приведены входные данные программы с подробным описанием типа данных, его размера, диапазона изменений, точности представления и формы ввода.

Таблица 3 – Входные данные

№	Имя (физический смысл)	Тип	Размер в байтах	Диапазон изменений	Точность представления	Форма ввода
1	2	3	4	5	6	7
1	Email пользователя	String	100	От А до Z, от 0 до 9, символы @ .	Текст	Ввод с клавиатуры
2	Пароль пользователя	String	100	От А до Z, от 0 до 9, спецсимволы	Текст	Ввод с клавиатуры
3	Имя пользователя	String	50	От А до Z, кириллица	Текст	Ввод с клавиатуры
4	Дата рождения	Date	8	Формат ДД.ММ.ГГГГ	Дата	Выбор даты
5	Номер телефона	String	20	Формат +7XXXXXXXXXX	Текст	Ввод с клавиатуры
6	Повтор пароля	String	100	От А до Z, от 0 до 9, спецсимволы	Текст	Ввод с клавиатуры

№	Имя (физический смысл)	Тип	Размер в байтах	Диапазон изменений	Точность представления	Форма ввода
1	2	3	4	5	6	7
7	VIN автомобиля	String	30	17 символов (латиница + цифры)	Текст	Ввод с клавиатуры
8	Марка автомобиля	String	50	Название марки	Текст	Выбор из выпадающего списка
9	Модель автомобиля	String	50	Название модели	Текст	Выбор из выпадающего списка
10	Год выпуска автомобиля	Int	4	От 1900 до текущего года	Целое число	Ввод с клавиатуры
11	Цвет автомобиля	String	30	Название цвета	Текст	Выбор из выпадающего списка
12	Объём двигателя	Double	8	От 0.5 до 8.0 литров	С плавающей точкой	Ввод с клавиатуры
13	Мощность двигателя	Int	4	От 30 до 1000 л.с.	Целое число	Ввод с клавиатуры
14	Пробег автомобиля	Int	4	От 0 до 500000 км	Целое число	Ввод с клавиатуры
15	Количество владельцев	Int	4	От 1 до 10	Целое число	Ввод с клавиатуры
16	Цена автомобиля	Int	4	От 10000 до 50000000 Р	Целое число	Ввод с клавиатуры
17	Описание объявления	String	1000	Свободный текст до 1000 символов	Текст	Ввод с клавиатуры
18	Город продажи	String	50	Название города	Текст	Выбор из выпадающего списка
19	Тип кузова	String	30	Седан, хэтчбек, кроссовер и т.д.	Текст	Выбор из выпадающего списка
20	Тип трансмиссии	String	30	АКПП, МКПП и др.	Текст	Выбор из выпадающего списка
21	Тип топлива	String	30	Бензин, дизель, электро и т.д.	Текст	Выбор из выпадающего списка
22	Тип привода	String	30	Передний, задний, полный	Текст	Выбор из выпадающего списка
23	Изображение автомобиля	Bitmap	500 000+	До 5 файлов	Битовая карта	Выбор из галереи устройства
24	Заголовок записи в бортжурнале	String	100	Свободный текст	Текст	Ввод с клавиатуры

№	Имя (физический смысл)	Тип	Размер в байтах	Диапазон изменений	Точность представления	Форма ввода
1	2	3	4	5	6	7
25	Текст записи в бортжурнале	String	1000	Свободный текст	Текст	Ввод с клавиатуры
26	Пробег записи бортжурнала	Int	4	От 0 до 500000 км	Целое число	Ввод с клавиатуры
27	Новая марка авто (для админа)	String	50	Название марки	Текст	Ввод с клавиатуры
28	Новая модель авто (для админа)	String	50	Название модели	Текст	Ввод с клавиатуры

В Таблице 5 приведены выходные данные программы.

Таблица 4 – Выходные данные

№	Имя (физический смысл)	Тип	Размер в байтах	Диапазон изменений	Точность представления	Форма вывода
1	2	3	4	5	6	7
Таблица пользователей (User)						
1	Name	String	40	Текстовое поле	от А до z, 0-9, символы	Вывод в текстовое поле профиля
2	Email	String	40	Текстовое поле	от А до z, 0-9, символы	Вывод в текстовое поле профиля
3	Phone	String	20	Текстовое поле	только цифры	Вывод в текстовое поле профиля
4	Password (зашифрованный)	String	64	Хэширование	Строка хэш-значений	Внутреннее хранение
5	ProfileImage	String (Base64)	50000	Текстовое поле (Base64)	Двоичные данные	Вывод изображения в ImageView профиля
Таблица объявлений (Car)						
6	Brand	String	40	Текстовое поле	от А до z, 0-9, символы	Вывод на карточке и в деталях
7	Model	String	40	Текстовое поле	от А до z, 0-9, символы	Вывод на карточке и в деталях
8	Price	Long	8	Числовое поле	Целое число	Вывод на карточке
9	Mileage	Long	8	Числовое поле	Целое число	Вывод на карточке

№	Имя (физический смысл)	Тип	Размер в байтах	Диапазон изменений	Точность представления	Форма вывода
1	2	3	4	5	6	7
10	Year	Int	4	Числовое поле	Целое число	Вывод в характеристиках
11	Volume	Double	8	Числовое поле	Дробное число	Вывод в характеристиках
12	Horsepower	Int	4	Числовое поле	Целое число	Вывод в характеристиках
13	Drive	String	20	Текстовое поле	от А до z	Вывод в характеристиках
14	Transmission	String	20	Текстовое поле	от А до z	Вывод в характеристиках
15	Fuel	String	20	Текстовое поле	от А до z	Вывод в характеристиках
16	Color	String	20	Текстовое поле	от А до z	Вывод в характеристиках
17	BodyType	String	20	Текстовое поле	от А до z	Вывод в характеристиках
18	City	String	30	Текстовое поле	от А до z	Вывод в карточке объявления
19	Owners	Int	4	Числовое поле	Целое число	Вывод в характеристиках
20	Description	String	500	Текстовое поле	Символьное поле	Вывод в описании объявления
21	PhotoOne	String (Base64)	50000	Текстовое поле	Base64 код изображения	Вывод изображения
22	PhotoTwo	String (Base64)	50000	Текстовое поле	Base64 код изображения	Вывод изображения
23	PhotoThree	String (Base64)	50000	Текстовое поле	Base64 код изображения	Вывод изображения
24	PhotoFour	String (Base64)	50000	Текстовое поле	Base64 код изображения	Вывод изображения
25	PhotoFive	String (Base64)	50000	Текстовое поле	Base64 код изображения	Вывод изображения
Таблица избранного (Favorites)						
26	FavoriteCarIds	Array of String	500	Массив строк	ID объявлений	Использование при отображении избранного
Таблица боржурнала (Bortjournal)						
27	Title	String	100	Текстовое поле	от А до z, 0-9, символы	Вывод в карточке записи
28	Text	String	1000	Текстовое поле	Символьное поле	Вывод в деталях записи
29	Mileage	Long	8	Числовое поле	Целое число	Вывод в деталях записи

№	Имя (физический смысл)	Тип	Размер в байтах	Диапазон изменений	Точность представления	Форма вывода
1	2	3	4	5	6	7
30	Brand	String	40	Текстовое поле	от А до z, 0-9, символы	Вывод в карточке записи
31	Model	String	40	Текстовое поле	от А до z, 0-9, символы	Вывод в карточке записи

2.2.3. Методы

В ходе разработки мобильного приложения "Drivee" были использованы следующие методы программирования:

Объектно-Ориентированное Программирование (ООП):

Программа построена на принципах ООП, где каждый элемент системы (например, пользователь, автомобиль, объявление) представлен в виде объектов с собственными полями и методами. Это обеспечило модульность кода, упрощение его сопровождения и расширения.

Структурное программирование:

При реализации отдельных модулей (например, регистрация пользователя, фильтрация объявлений, работа с профилем) применялись принципы структурного программирования. Логика работы приложения была разделена на последовательные, понятные блоки, что повысило читаемость и устойчивость кода.

Реактивное программирование:

При взаимодействии с базой данных Firebase (Firestore) и обработки событий пользовательского интерфейса применялись элементы реактивного подхода. Изменения данных мгновенно отражаются в интерфейсе без необходимости ручного обновления — например, при добавлении новой машины в список объявлений или изменении профиля пользователя.

2.2.4. Тесты

Сценарий тестирования и результаты тестовых испытаний приведен в Приложении В. Сценарий тестов и результаты тестовых испытаний.

2.2.5. Контроль целостности данных

В Таблице 5 представлен контроль целостности данных в приложении "Drivee", описывающий типичные ситуации и реакцию системы при сохранении, выводе, изменении или удалении данных.

Таблица 5 – Контроль целостности данных

№	Ситуация	Аномалия	Реакция программы
1	2	3	4
1	Попытка регистрации с уже существующим email	Пользователь уже существует в базе данных	Отображение сообщения "Аккаунт с таким email уже существует" и запрет на регистрацию
2	Попытка создать объявление без заполнения всех обязательных полей	Некорректный объект в базе данных	Появление предупреждения "Заполните все обязательные поля" и запрет на отправку формы
3	Изменение информации в объявлении (например, цены)	Мгновенное обновление данных без подтверждения	Изменения сохраняются только после явного нажатия кнопки "Сохранить"
4	Попытка удалить объявление без подтверждения	Ошибочное удаление объявления	Появляется диалог подтверждения "Вы уверены, что хотите удалить объявление?"
5	Добавление машины в "Избранное", потом её удаление из базы администратором	Ссылка в "Избранном" на несуществующий объект	При попытке открытия карточки - отображение сообщения "Объявление больше недоступно" и удаление его из списка
6	Изменение профиля пользователя без подключения к интернету	Данные не сохраняются в базе данных	Выводится сообщение об ошибке "Нет подключения к интернету. Повторите попытку позже."
7	Оформление заказа без выбора машины	Некорректный заказ в базе	Блокировка кнопки "Оформить заказ", пока не будет добавлено хотя бы одно объявление

2.3. Проектирование

2.3.1. Схема архитектуры приложения

Клиент отправляет запрос в облачное хранилище через Web-API Firebase SDK. Этот SDK, находясь внутри облачного хранилища, обрабатывает запрос и взаимодействует с базой данных Firebase Firestore. После того как база данных возвращает нужную информацию, SDK формирует ответ и отправляет его обратно клиенту.

На Рисунке 16 представлена клиент-серверная архитектура программы.

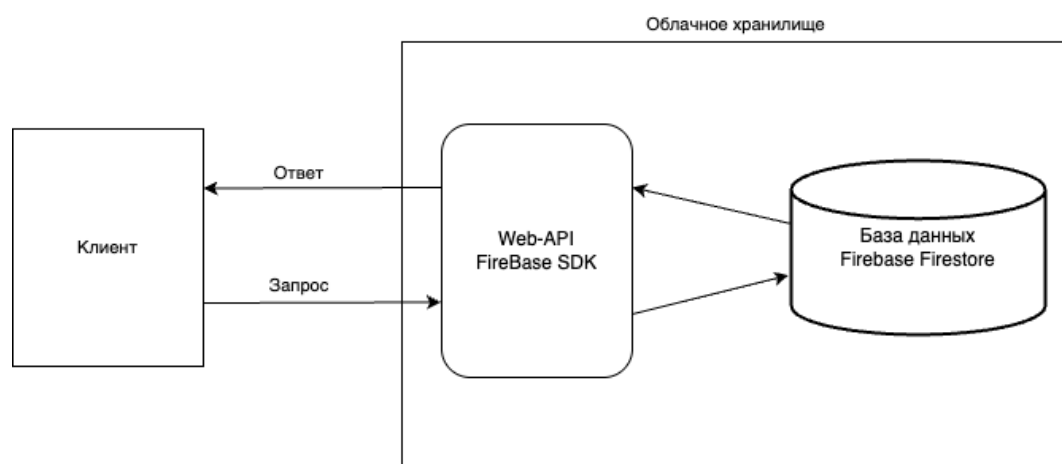


Рисунок 16 - Архитектурная схема приложения

2.3.2. Логическая схема данных

Разработка данной схемы не предусмотрена в связи с отсутствием необходимости в рамках поставленных задач.

2.3.3. Физическая схема данных

Разработка данной схемы не предусмотрена в связи с отсутствием необходимости в рамках поставленных задач.

2.3.4. Структурная схема

На Рисунке 17-19 представлена структурная схема программы. Большинство классов в программе являются глобальными и выступают как вспомогательные элементы, поэтому на схеме приведены зависимые файлы проекта.

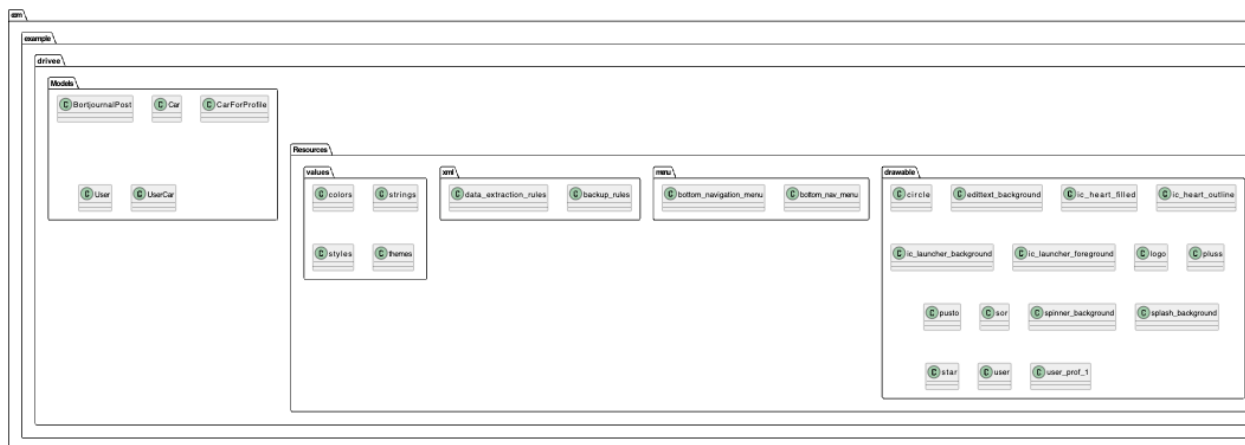


Рисунок 17 – Структурная схема приложения часть 1

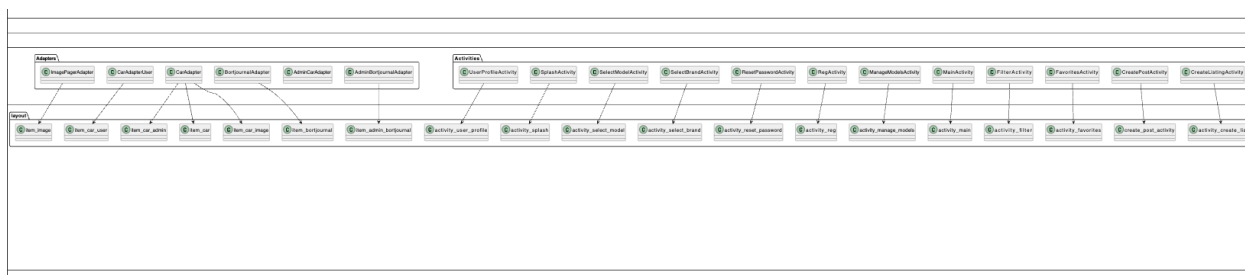


Рисунок 18 - Структурная схема приложения часть 2

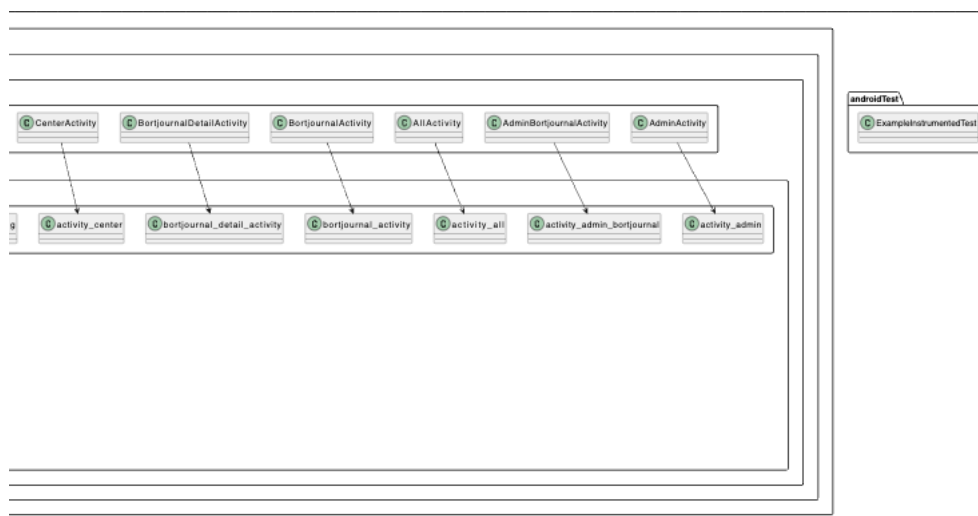


Рисунок 19 - Структурная схема приложения часть 1

2.3.5. Функциональная схема

На диаграмме изображена структура мобильного приложения для продажи и покупки автомобилей. Она состоит из различных страниц, каждая из которых выполняет определённую функцию и взаимодействует с другими. Вот простое описание:

Приложение начинается с экранов входа, регистрации и сброса пароля, которые позволяют пользователю авторизоваться или создать новый аккаунт.

После успешного входа пользователь попадает на главную страницу, где можно перейти к фильтрации объявлений, просмотру профиля, созданию записи, или добавлению объявлений в избранное.

Есть отдельные страницы для создания объявлений и создания записей, где вводятся данные об автомобиле (тип кузова, топливо, привод, марка, модель, VIN, год, цена и пр.). Также присутствует автоматическое определение города по геолокации.

Избранное и профиль пользователя позволяют управлять сохранёнными объявлениями и личной информацией. Страницы подробной информации об объявлениях и записях предоставляют подробности и возможность перейти к другим частям приложения.

Отдельно выделены страницы администрирования, где можно управлять объявлениями, записями, а также добавлять марки и модели автомобилей.

Все страницы связаны между собой переходами, что обеспечивает логичную и полную навигацию по приложению.

На Рисунке 20 представлена функциональная схема.

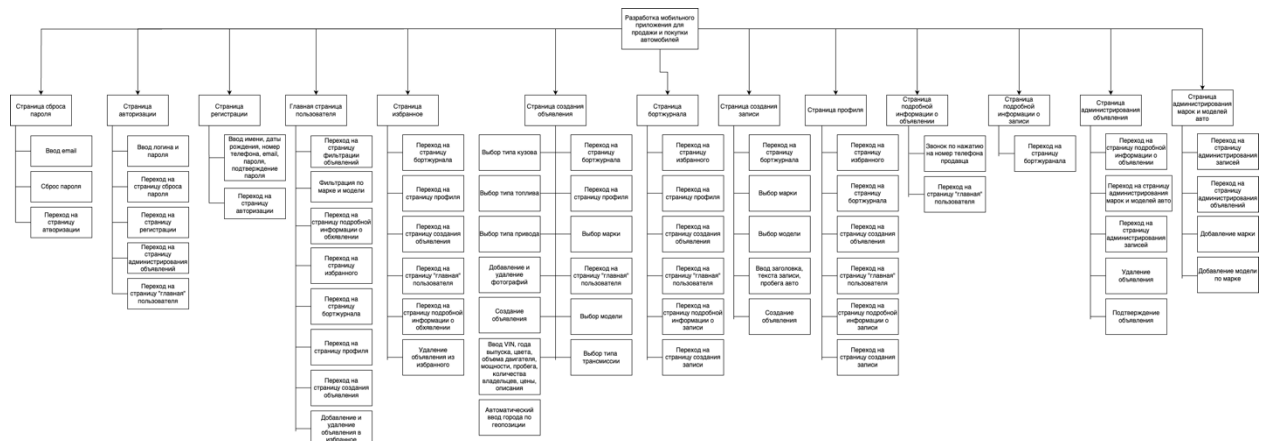


Рисунок 20 - Функциональная схема

2.3.6. Диаграмма классов

На Рисунках 21-23 представлены части диаграммы классов программы. В классах указаны внутренние атрибуты и методы, используемые программой.

Связь между классами показана стрелками. В глобальных элементах и интерфейсах стрелки не присутствуют, однако также описаны внутренние члены классов.

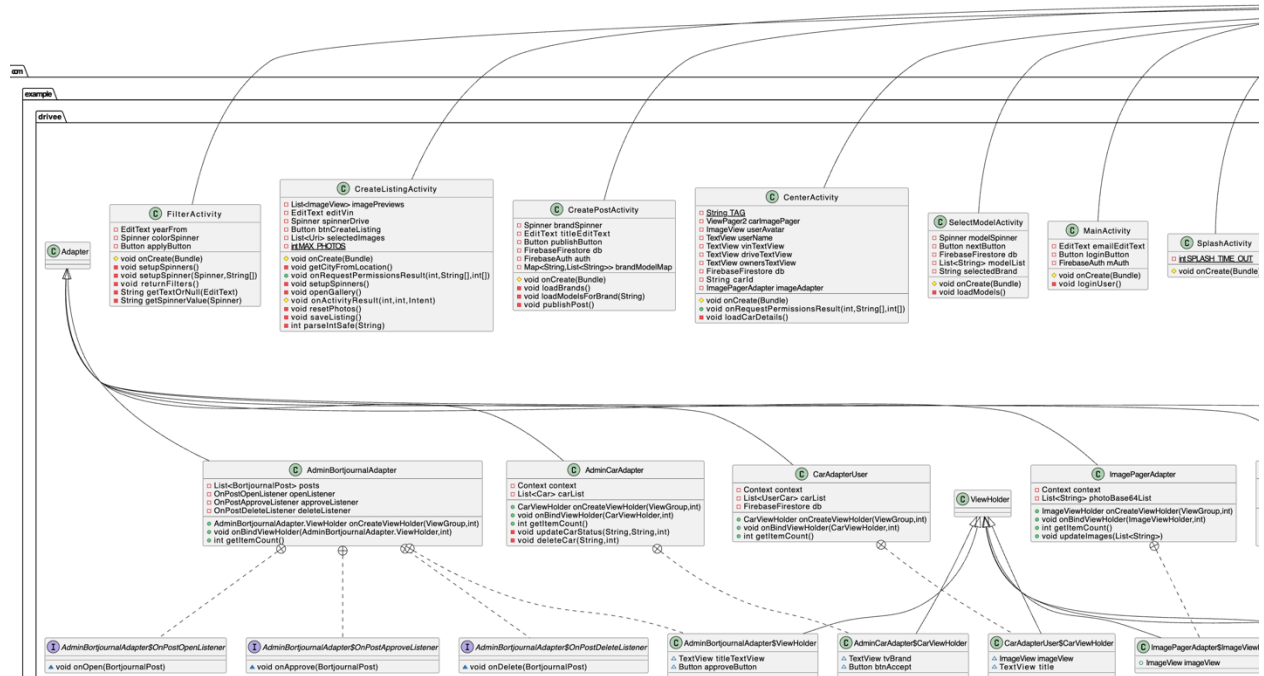


Рисунок 21 - Диаграмма классов часть 1

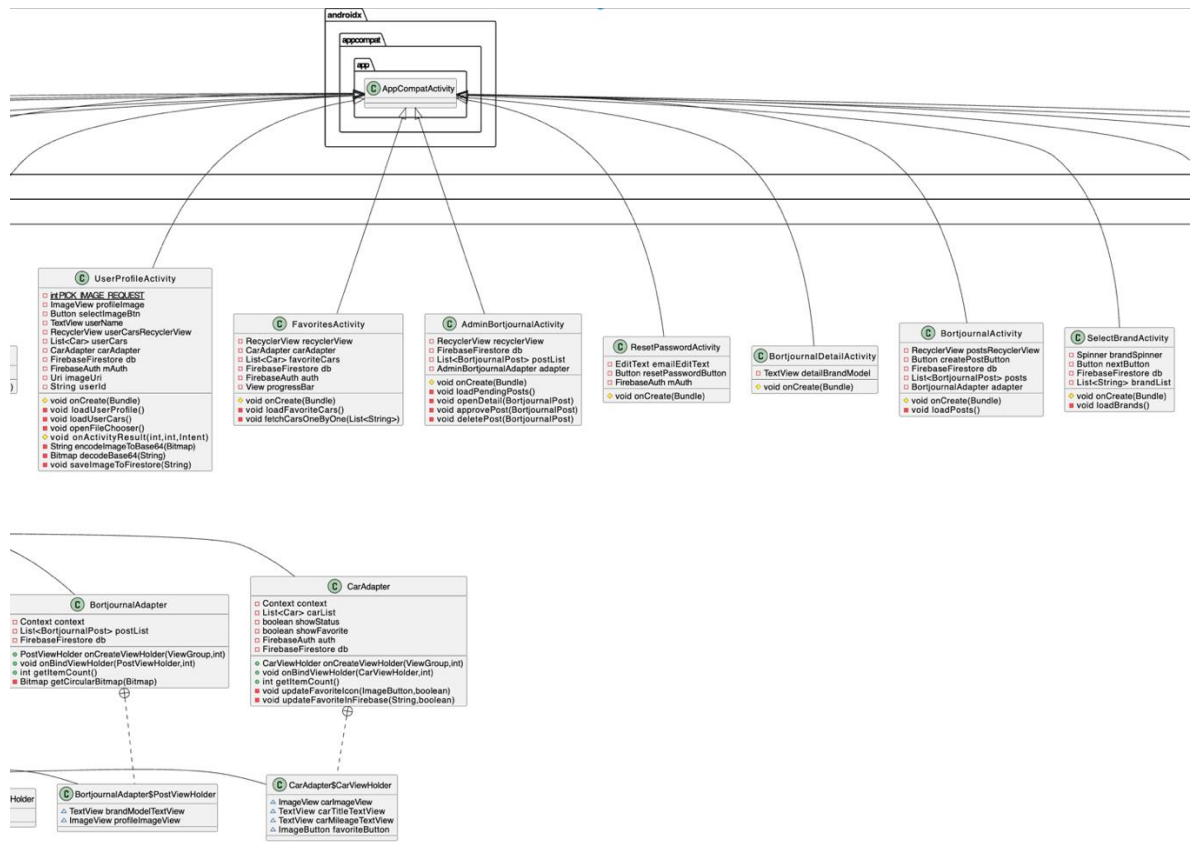


Рисунок 22 - Диаграмма классов часть 2

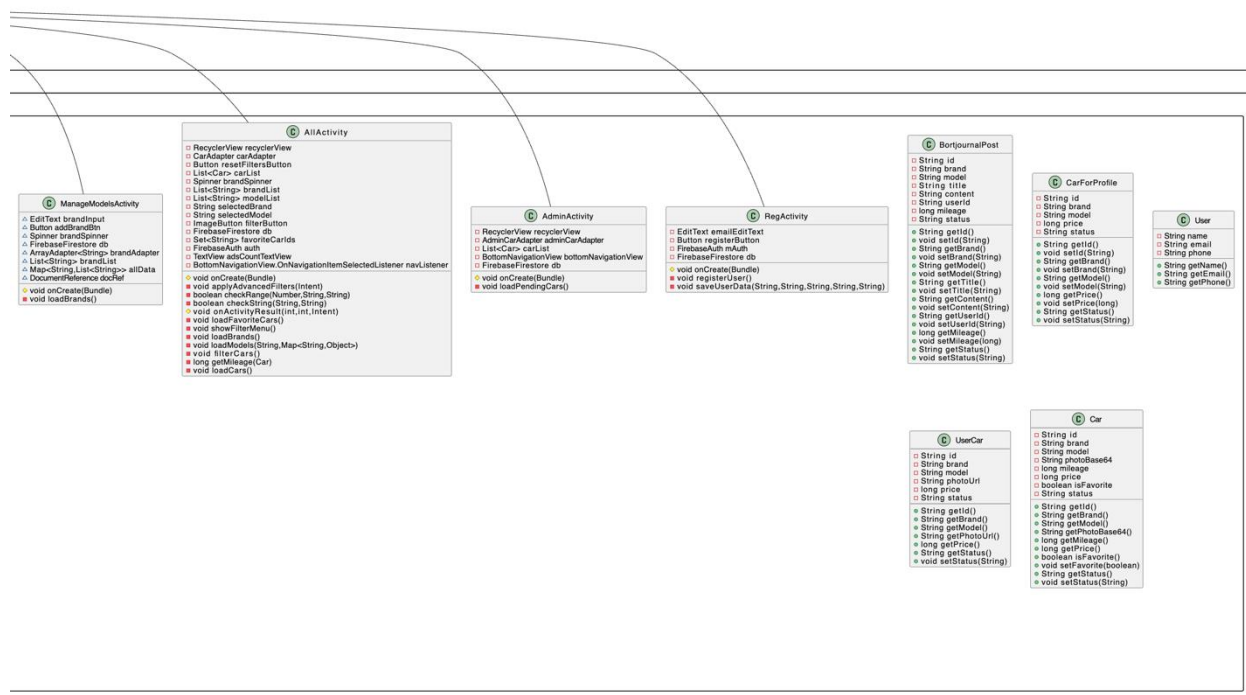


Рисунок 23 - Диаграмма классов часть 3

Таблица 6 – Описание диаграммы классов

№	Название	Описание
1	AndroidManifest.xml	конфигурация приложения: разрешения, иконки, список активити
2	AdminActivity.java	панель администратора, просмотр необработанных объявлений
3	AdminBortjournalActivity.java	панель администратора для управления бортжурналами
4	AdminBortjournalAdapter.java	адаптер для отображения бортжурналов в админке
5	AdminCarAdapter.java	адаптер для списка машин в админке.
6	AllActivity.java	главный экран со всеми объявлениями + фильтры и навигация.
7	BortjournalActivity.java	экран бортжурнала пользователя
8	BortjournalAdapter.java	адаптер для списка записей в бортжурнале
9	BortjournalDetailActivity.java	экран подробностей по записи в бортжурнале
10	BortjournalPost.java	модель (POJO) для записи в бортжурнал
11	Car.java	модель машины
12	CarAdapter.java	адаптер для отображения карточек машин
13	CarAdapterUser.java	адаптер для отображения машин текущего пользователя
14	CarForProfile.java	отдельная модель машины для профиля
15	CenterActivity.java	экран карточки машины
16	CreateListingActivity.java	создание нового объявления
17	CreatePostActivity.java	создание записи в бортжурнал

№	Название	Описание
1	2	3
18	FavoritesActivity.java	избранные объявления пользователя
№	Название	Описание
19	FilterActivity.java	расширенные фильтры поиска
20	ImagePagerAdapter.java	адаптер для ViewPager с фото машины
21	MainActivity.java	экран авторизации
22	ManageModelsActivity.java	админка: управление марками и моделями
23	RegActivity.java	экран регистрации
24	ResetPasswordActivity.java	сброс пароля
25	SelectBrandActivity.java	выбор марки авто.
26	SelectModelActivity.java	выбор модели авто.
27	SplashActivity.java	splash-экран при запуске
28	User.java	модель пользователя
29	UserCar.java	модель машины, привязанной к пользователю
30	UserProfileActivity.java	профиль пользователя

2.3.7. Схема тестирования

Функциональное тестирование охватывает базовые функции приложения: авторизация, регистрация, сброс пароля, загрузка профиля пользователя, выход из аккаунта, просмотр и администрирование объявлений и записей, создание объявлений и записей, применение фильтров поиска, работа с избранным, а также управление марками и моделями авто.

Тестирование производительности проверяет скорость загрузки различных страниц (авторизация, главная, профиль, бордюркула, создание объявления), а также работу приложения при одновременной активности нескольких пользователей.

Тестирование пользовательского интерфейса включает проверку отображения карточек машин, фото профиля, выпадающих списков, форм создания объявлений, навигации и адаптации интерфейса под разные размеры экранов.

Тестирование кроссплатформенности фокусируется на работоспособности приложения на разных версиях Android (от 8 до 12) и на различных устройствах.

Вся схема систематизирует процесс тестирования, чтобы обеспечить надёжность и удобство использования приложения.

На Рисунке 24 представлена схема тестирования настольного приложения «Drivee».

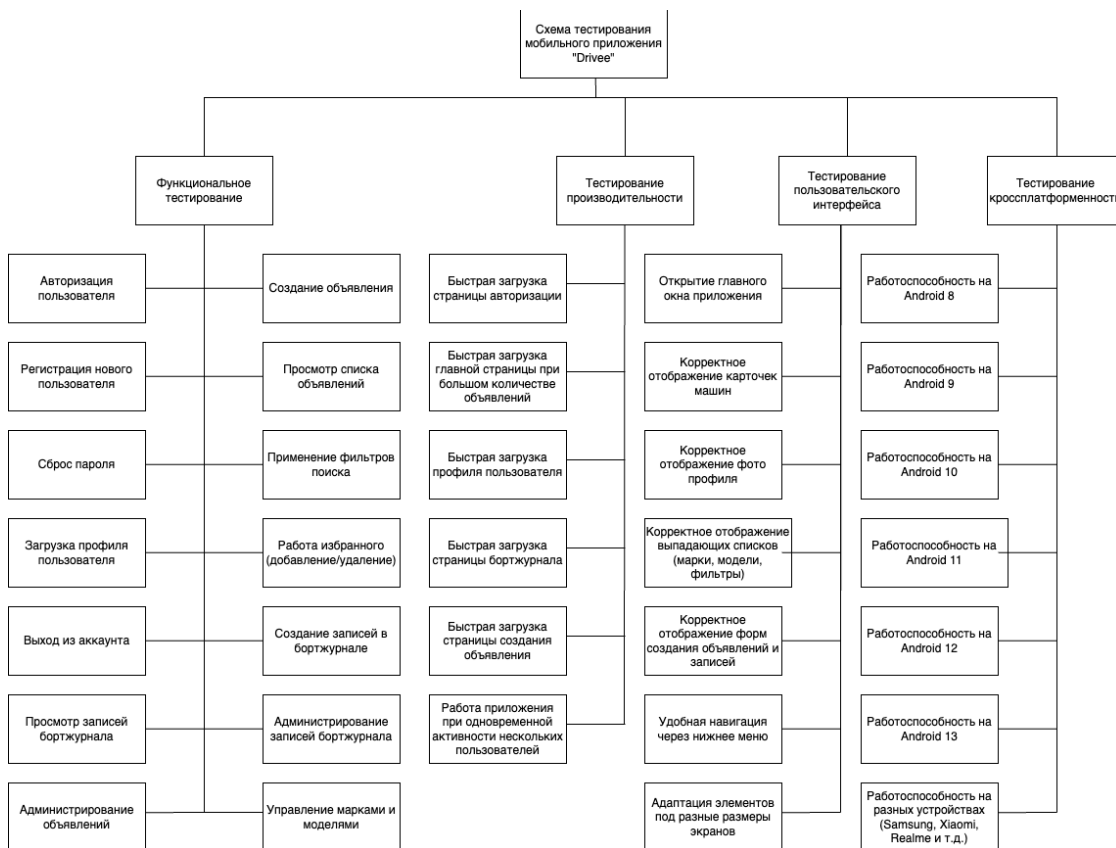


Рисунок 24 – Схема тестирования

2.3.8. Схема пользовательского интерфейса

Схема показывает, как пользователь перемещается между основными экранами мобильного приложения. Всё начинается со страницы авторизации, откуда можно попасть на регистрацию или сброс пароля. После входа открывается главная страница пользователя с доступом к профилю, поиску, боржурналу, избранному и созданию объявлений. Из этих разделов можно перейти на страницы выбора марки и модели автомобиля, а также на подробную информацию о машине. Для администратора предусмотрена отдельная ветка с доступом к модерации, управлению боржурналом и редактированию марок и моделей. Всё соединено так, чтобы переходы между функциями были логичными и быстрыми.

На Рисунке 25 представлена схема пользовательского интерфейса.

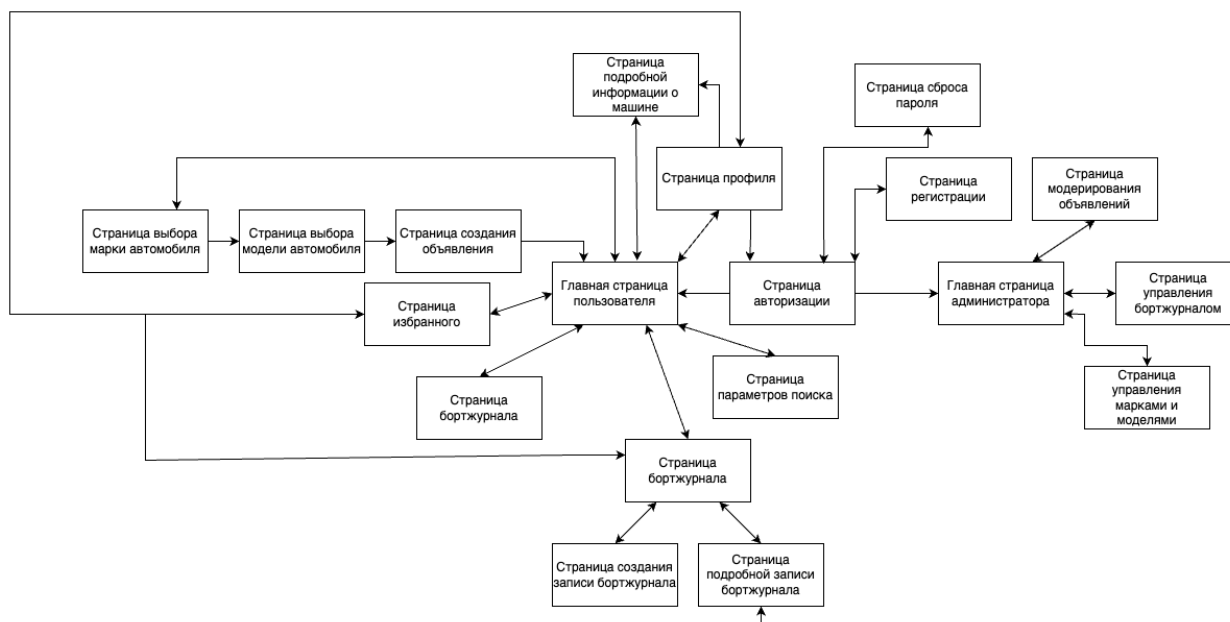


Рисунок 25 – Схема пользовательского интерфейса

2.4. Результат работы программы

На Рисунках 26-37 кратко продемонстрированы работы программы.

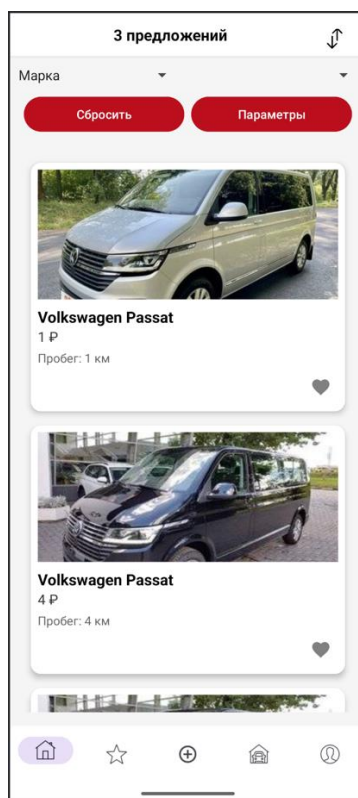


Рисунок 26 – Главная страница пользователя

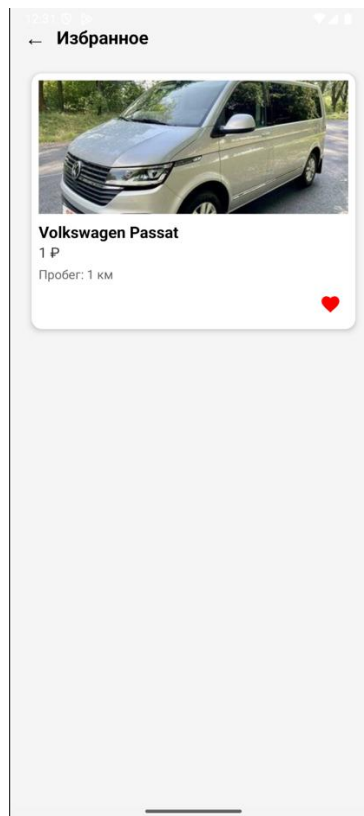


Рисунок 27 – Страница избранное

A screenshot of a mobile application interface for creating a car listing. At the top, there are two gray boxes with white photo icons. Below them is a black button labeled "Загрузить фото" and a red button labeled "Сбросить". The form consists of several input fields: "VIN", "Volkswagen", "Golf", "Год выпуска", "Топливо" (with a dropdown arrow), "Цвет", "Тип кузова" (with a dropdown arrow), "Привод" (with a dropdown arrow), "Коробка" (with a dropdown arrow), and "Объем двигателя (л)". The background is a light gray.

Рисунок 28 – Страница создания объявления

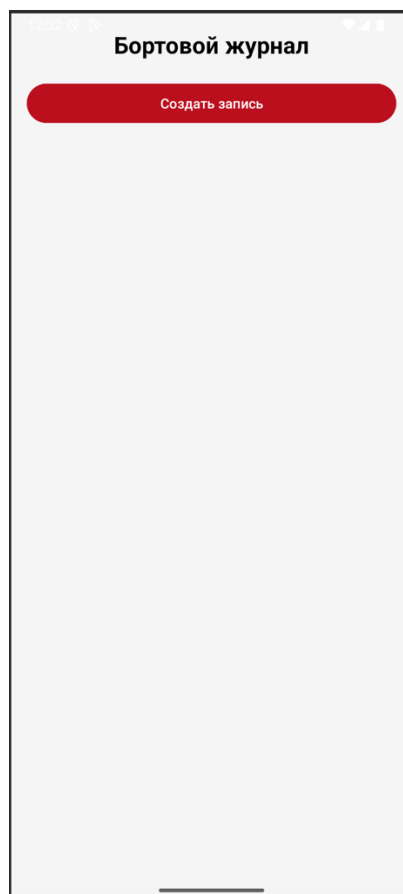


Рисунок 29 – Страница бортового журнала

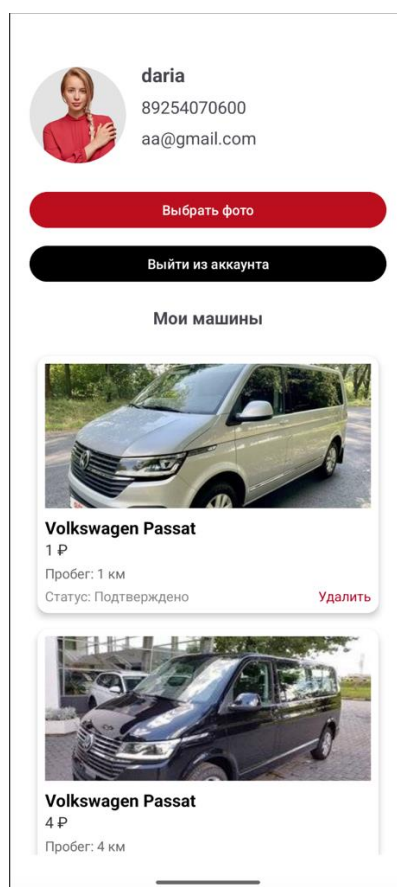


Рисунок 30 – Страница пользователя



Рисунок 31 – Подробная информации об объявлении

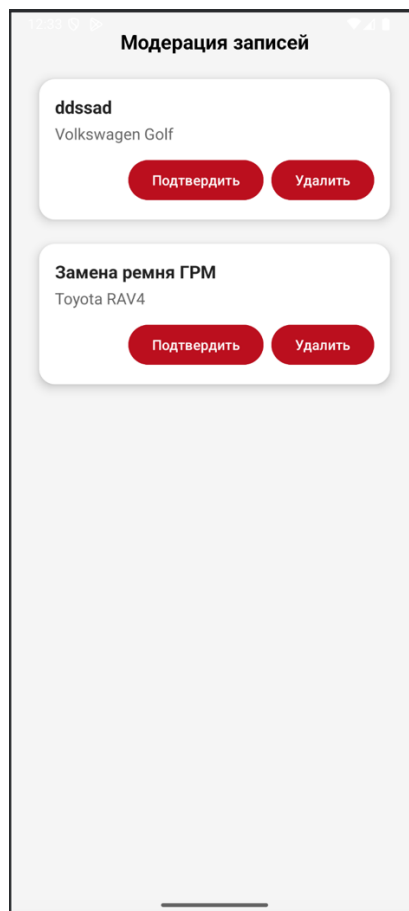


Рисунок 32 – Модерирование записей

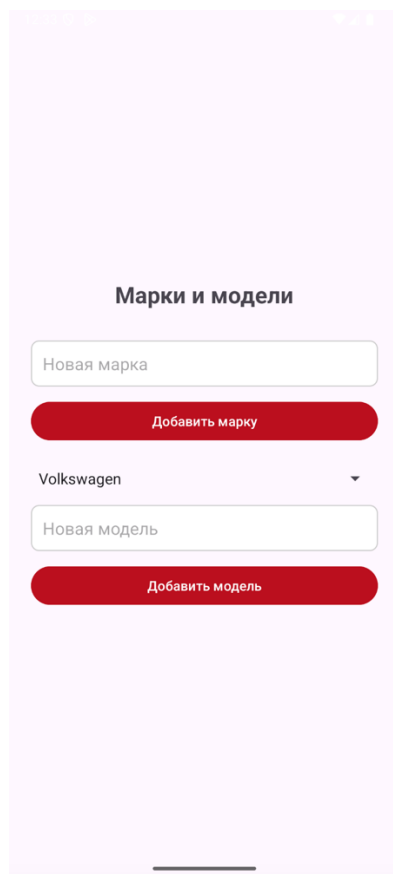


Рисунок 33 – Страница редактирования марок и моделей авто

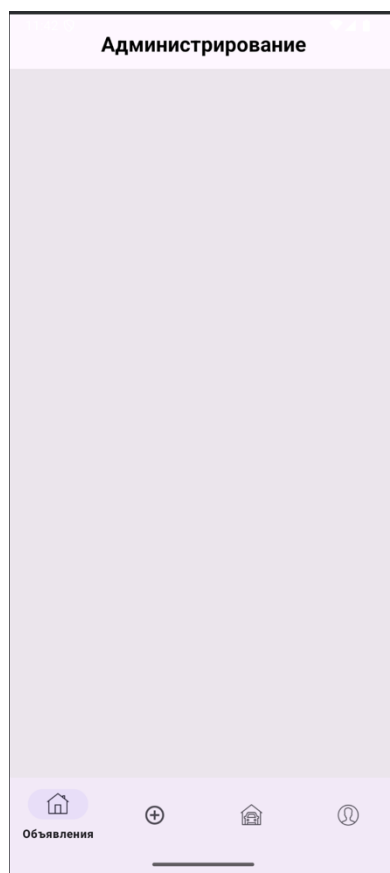


Рисунок 34 – Администрирование объявлений



The login screen for the DRIVVY application. At the top is the DRIVVY logo, which consists of a red car icon and the word "DRIVVY" in bold black letters. Below the logo is the title "Вход" (Login). There are two input fields: "Email" and "Пароль" (Password). Below these fields are two red buttons: "Войти" (Login) and "Зарегистрироваться" (Register). At the bottom, there is a link "Сбросить пароль" (Reset password).

DRIVVY

Вход

Email

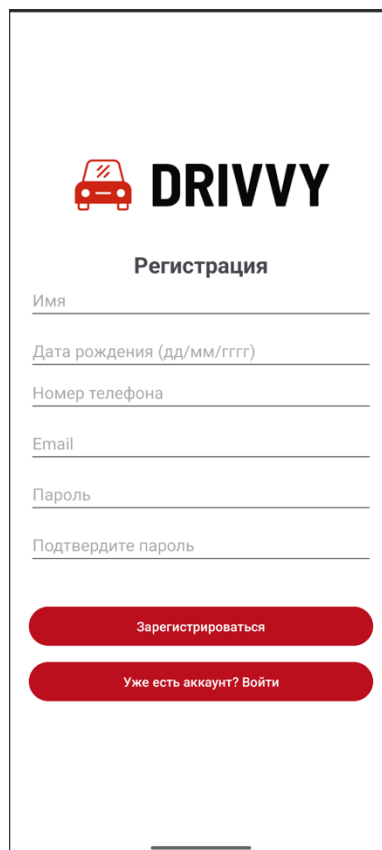
Пароль

Войти

Зарегистрироваться

Сбросить пароль

Рисунок 35 – Страница атверизации



The registration screen for the DRIVVY application. At the top is the DRIVVY logo, which consists of a red car icon and the word "DRIVVY" in bold black letters. Below the logo is the title "Регистрация" (Registration). There are six input fields: "Имя" (Name), "Дата рождения (дд/мм/гггг)" (Date of birth), "Номер телефона" (Phone number), "Email", "Пароль" (Password), and "Подтвердите пароль" (Confirm password). Below these fields are two red buttons: "Зарегистрироваться" (Register) and "Уже есть аккаунт? Войти" (Already have an account? Login).

DRIVVY

Регистрация

Имя

Дата рождения (дд/мм/гггг)

Номер телефона

Email

Пароль

Подтвердите пароль

Зарегистрироваться

Уже есть аккаунт? Войти

Рисунок 36 – Страница регистрации

Сброс пароля

Введите ваш Email

Сбросить пароль

Помните пароль? Войти

Рисунок 37 – Страница сброса пароля

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1. Инструментальные средства

Для хранения данных в мобильном приложении Drivee используется облачная база данных Firebase Firestore, которая предоставляет высокую производительность, масштабируемость и поддержку работы в реальном времени. Это решение обеспечивает надежное хранение информации о пользователях, объявлениях, избранных автомобилях, борджурналах и другой сопутствующей информации.

В качестве среды разработки использовалась Android Studio, являющаяся стандартной IDE для создания мобильных приложений на платформе Android. Программная логика приложения реализована на языке Java, что позволило добиться стабильности, гибкости и удобной интеграции с Firebase-сервисами.

Разработка пользовательского интерфейса велась с использованием языка разметки XML, применяемого для создания всех экранов приложения: авторизации, регистрации, главной страницы, профиля пользователя, борджурнала, страниц создания объявлений и фильтрации.

Для реализации работы с изображениями автомобилей и профилей применялись инструменты работы с Base64 кодированием, а для отображения изображений внутри элементов управления — адаптеры RecyclerView и ViewPager2.

Все сетевые взаимодействия, процессы авторизации, регистрации, восстановления пароля, создания объявлений, а также управление профилем были организованы через стандартные библиотеки Android SDK и интеграцию с Firebase Authentication, Firebase Firestore и Firebase Storage.

3.2. Отладка программы

Отладка программы проводилась с использованием встроенных инструментов отладки, предоставляемых средой разработки Android Studio. Процесс отладки осуществлялся в режиме реального времени на эмуляторе

устройства Android и на реальных устройствах, что позволило эффективно отслеживать ошибки в логике приложения и корректировать их.

Основными типами ошибок в процессе тестирования являлись:

Ошибки привязки данных к элементам пользовательского интерфейса из-за неправильного указания идентификаторов id в файлах разметки XML;

Ошибки взаимодействия с базой данных Firebase, такие как неправильная структура запросов к коллекциям или попытки чтения несуществующих полей документов;

Ошибки обработки изображений, связанные с кодированием и декодированием изображений в формате Base64;

Ошибки навигации между активностями, вызванные неправильной передачей данных через Intent;

Ошибки валидации данных при регистрации, авторизации и создании объявлений.

Особое внимание при отладке уделялось проверке корректности выполнения запросов к серверу Firebase:

Чтение данных из коллекций объявлений и пользователей;

Сохранение новых документов в базе данных;

Удаление объектов (объявлений, избранных машин, записей в бортжурнале);

Изменение данных профиля пользователя и объявлений.

Дополнительно выявленные ошибки сериализации объектов и несоответствия моделей данных были устранены путем согласования структуры моделей на стороне приложения с фактической структурой коллекций в Firebase.

3.3. Защитное программирование

При разработке мобильного приложения Drivee были реализованы принципы защитного программирования для повышения устойчивости системы к некорректным действиям пользователей и сбоям в процессе работы.

Меры защитного программирования включали:

Проверка всех вводимых данных на соответствие требованиям:

Ввод электронной почты проверяется на корректность формата.

Поля пароля проверяются на минимальную длину и наличие обязательных символов (цифры, буквы в разных регистрах).

Проверка на совпадение пароля и подтверждения пароля при регистрации.

При создании объявления все текстовые поля проверяются на заполненность, числовые поля — на корректность числового ввода.

Обработка ошибок сетевого взаимодействия с Firebase:

При потере подключения к интернету пользователю выводятся понятные сообщения об ошибке.

Повторная попытка выполнения операций при неудачных сетевых запросах.

Обработка возможных ошибок загрузки изображений:

Проверка наличия выбранного изображения перед его загрузкой.

Уведомление пользователя в случае неудачной загрузки или повреждения файла.

Защита от некорректной навигации:

Все переходы между активностями сопровождаются проверкой передаваемых данных (Intent extras).

При отсутствии необходимых данных активность корректно завершается с выводом сообщения пользователю.

Проверка прав доступа:

Перед выполнением действий, требующих специальных разрешений (например, вызов по телефону), выполняется проверка наличия разрешения.

При отсутствии разрешения запрашивается его предоставление у пользователя.

Обработка исключительных ситуаций (try-catch):

Основные операции взаимодействия с базой данных, обработка изображений и операций с файлами обернуты в блоки try-catch для перехвата возможных исключений без аварийного завершения приложения.

Применение данных принципов позволило значительно повысить надежность работы приложения, минимизировать количество аварийных завершений и обеспечить удобство для пользователей даже в непредвиденных ситуациях.

3.4. Характеристики программы

В Таблице 7 представлена характеристика программы в виде перечисления всех модулей.

Таблица 7 - Характеристика программы

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
1	AndroidManifest.xml	<code><manifest xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools"> <uses-feature android:name="android.hardware.telephony" android:required="false" /> <uses-permission android:name="android.permission.CALL_PHONE" /> <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" /> <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" /></code>	2 Кб	Нет
2	AdminActivity.java	<code>import android.content.Intent; import android.os.Bundle; import android.util.Log; import androidx.appcompat.app.AppCompatActivity;</code>	5 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView; import com.google.android.material.bottomnavigation.BottomNavigationView; import com.google.firebase.firestore.FirebaseFirestore; import com.google.firebase.firestore.QueryDocumentSnapshot; import java.util.ArrayList; import java.util.List;		
3	AdminBortjournalActivity.java	import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Toast; import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView; import com.google.firebase.firestore.FirebaseFirestore; import com.google.firebase.firestore.QueryDocumentSnapshot; import java.util.ArrayList; import java.util.List;	4 Кб	Нет
4	AdminBortjournalAdapter.java	import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.Button; import android.widget.TextView;	5 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView; import java.util.List;		
5	AdminCarAdapter.java	import android.content.Context; import android.content.Intent; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.Button; import android.widget.TextView; import android.widget.Toast; import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView; import com.google.firebase.firestore.FirebaseFirestore; import java.util.List;	6 Кб	Нет
6	AllActivity.java	import android.content.Intent; import android.os.Bundle; import android.util.Log; import android.view.MenuItem; import android.view.View; import android.widget.AdapterView; import android.widget.Button; import android.widget.ImageButton; import android.widget.Spinner; import android.widget.TextView; import android.widget.Toast; import androidx.annotation.NonNull; import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager;	9 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import androidx.recyclerview.widget.RecyclerView; import com.google.android.material.bottomnavigation.BottomNavigationView; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.firestore.FirebaseFirestore; import com.google.firebase.firestore.QueryDocumentSnapshot; import java.util.ArrayList; import java.util.HashSet; import java.util.List; import java.util.Map; import java.util.Set;		
7	BortjournalActivity.java	import android.content.Intent; import android.os.Bundle; import android.widget.Button; import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView; import com.google.firebase.firestore.FirebaseFirestore; import com.google.firebase.firestore.Query; import com.google.firebase.firestore.QueryDocumentSnapshot; import java.util.ArrayList; import java.util.List;	2 Кб	Нет
8	BortjournalAdapter.java	import android.content.Context; import android.content.Intent;	3 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.graphics.Canvas; import android.graphics.Paint; import android.graphics.PorterDuffXfer mode; import android.graphics.Rect; import android.graphics.RectF; import android.graphics.PorterDuff; import android.util.Base64; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ImageView; import android.widget.TextView; import androidx.annotation.NonNull; import androidx.recyclerview.widget.Rec yclerView; import com.google.firebase.firestore.Fire baseFirestore; import java.util.List;		
9	BortjournalDetailActivity.java	import android.os.Bundle; import android.widget.TextView; import androidx.appcompat.app.AppCom patActivity;	2 Кб	Нет
10	BortjournalPost.java	package com.example.drivее;	4 Кб	Нет
11	Car.java	import android.util.Log; import com.google.android.gms.common .api.internal.IStatusCallback;	2 Кб	Нет
12	CarAdapter.java	import android.content.Context; import android.content.Intent; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.util.Base64;	3 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ImageButton; import android.widget.ImageView; import android.widget.TextView; import android.widget.Toast; import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.firestore.FieldValue; import com.google.firebase.firestore.FirebaseFirestore; import java.text.DecimalFormat; import java.util.List;		
13	CarAdapterUser.java	import android.content.Context; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ImageView; import android.widget.TextView; import android.widget.Toast; import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView; import com.bumptech.glide.Glide; import com.google.firebase.firestore.FirebaseFirestore; import java.util.List;	3 Кб	Нет
14	CarForProfile.java	Нет	1 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
15	CenterActivity.java	import android.content.Intent; import android.content.pm.PackageManager; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.net.Uri; import android.os.Bundle; import android.util.Base64; import android.util.Log; import android.widget.ImageView; import android.widget.TextView; import android.widget.Toast; import androidx.appcompat.app.AppCompatActivity; import androidx.viewpager2.widget.View Pager2; import com.google.firebase.Timestamp; import com.google.firebase.firestore.Doc umentReference; import com.google.firebase.firestore.Doc umentSnapshot; import com.google.firebase.firestore.Fire baseFirestore; import java.text.SimpleDateFormat; import java.util.ArrayList; import java.util.Date; import java.util.List; import java.util.Locale;	4 Кб	Нет
16	CreateListingActivity.java	package com.example.drivее; import android.Manifest; import android.content.pm.PackageManager; import androidx.core.app.ActivityCompa t;	8 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import androidx.core.content.ContextCompat; import com.google.android.gms.location.FusedLocationProviderClient; import com.google.android.gms.location.LocationServices; import android.location.Address; import android.location.Geocoder; import java.util.Locale; import android.app.Activity; import android.content.Intent; import android.content.pm.PackageManager; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.location.Address; import android.location.Geocoder; import android.net.Uri; import android.os.Bundle; import android.util.Base64; import android.view.View; import android.widget.AdapterView; import android.widget.Button; import android.widget.EditText; import android.widget.ImageView; import android.widget.Spinner; import android.widget.Toast; import androidx.annotation.NonNull; import androidx.annotation.Nullable; import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat;		

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import androidx.core.content.ContextCompat; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.firestore.FirebaseFirestore; import java.io.ByteArrayOutputStream; import java.io.InputStream; import java.util.ArrayList; import java.util.HashMap; import java.util.List; import java.util.Locale; import java.util.Map;		
17	CreatePostActivity.java	import android.os.Bundle; import android.text.TextUtils; import android.widget.AdapterView; import android.widget.Button; import android.widget.EditText; import android.widget.Spinner; import android.widget.Toast; import androidx.annotation.Nullable; import androidx.appcompat.app.AppCompatActivity; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.firestore.FieldValue; import com.google.firebase.firestore.FirebaseFirestore; import java.util.ArrayList; import java.util.HashMap; import java.util.List; import java.util.Map;	3 Кб	Нет
18	FavoritesActivity.java	import android.content.Intent; import android.os.Bundle; import android.util.Log;	2 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import android.view.View; import android.widget.TextView; import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.firestore.FirebaseFirestore; import java.util.ArrayList; import java.util.List;		
19	FilterActivity.java	import android.content.Intent; import android.os.Bundle; import android.text.TextUtils; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.EditText; import android.widget.Spinner;	2 Кб	Нет
20	ImagePagerAdapter.java	import android.content.Context; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.util.Base64; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ImageView; import androidx.annotation.NonNull; import androidx.viewpager2.widget.ViewPager2; import androidx.recyclerview.widget.RecyclerView;	2 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
21	MainActivity.java	import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.EditText; import android.widget.Toast; import android.widget.Button; import androidx.appcompat.app.AppCompatActivity; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.auth.FirebaseUser; import com.google.firebase.firestore.DocumentSnapshot; import com.google.firebase.firestore.FirebaseFirestore;	2 Кб	Нет
22	ManageModelsActivity.java	import android.os.Bundle; import android.util.Log; import android.view.View; import android.widget.*; import androidx.appcompat.app.AppCompatActivity; import com.google.firebase.firestore.*; import java.util.*;	2 Кб	Нет
23	RegActivity.java	import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast; import androidx.appcompat.app.AppCompatActivity; import com.google.firebase.auth.FirebaseAuth; import com.google.firebase.auth.FirebaseUser;	3 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import com.google.firebase.firestore.FirebaseFirestore; import java.util.HashMap; import java.util.Map;		
24	ResetPasswordActivity.java	import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast; import androidx.appcompat.app.AppCompatActivity; import com.google.firebase.auth.FirebaseAuth;	1 Кб	Нет
25	SelectBrandActivity.java	import android.content.Intent; import android.os.Bundle; import android.util.Log; import android.view.View; import android.widget.AdapterView; import android.widget.Button; import android.widget.Spinner; import android.widget.Toast; import androidx.appcompat.app.AppCompatActivity; import com.google.firebase.firestore.DocumentSnapshot; import com.google.firebase.firestore.FirebaseFirestore; import java.util.ArrayList; import java.util.List; import java.util.Map;	2 Кб	Нет
26	SelectModelActivity.java	import android.content.Intent; import android.os.Bundle; import android.util.Log; import android.view.View; import android.widget.AdapterView; import android.widget.Button; import android.widget.Spinner; import android.widget.Toast;	2 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import androidx.appcompat.app.AppCompatActivity; import com.google.firebase.firestore.DocumentSnapshot; import com.google.firebase.firestore.FirebaseFirestore; import java.util.ArrayList; import java.util.List; import java.util.Map;		
27	SplashActivity.java	import android.content.Intent; import android.os.Bundle; import android.os.Handler; import androidx.appcompat.app.AppCompatActivity;	1 Кб	Нет
28	User.java	Нет	1 Кб	Нет
29	UserCar.java	Нет	1 Кб	Нет
30	UserProfileActivity.java	import android.content.Intent; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.net.Uri; import android.os.Bundle; import android.util.Base64; import android.util.Log; import android.widget.Button; import android.widget.ImageView; import android.widget.TextView; import android.widget.Toast; import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView; import com.google.firebase.auth.FirebaseAuth;	3 Кб	Нет

№	Название	Используемые библиотеки	Размер файла в Кб	Дополнительные файлы
1	2	3	4	5
		import com.google.firebase.firestore.FirebaseFirestore; import com.google.firebase.firestore.QueryDocumentSnapshot; import java.io.ByteArrayOutputStream; import java.io.FileNotFoundException; import java.io.InputStream; import java.util.ArrayList; import java.util.List;		

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта на тему «Разработка мобильного приложения "Drivee" для публикации и поиска автомобильных объявлений» было разработано мобильное приложение Drivee на языке программирования Java с использованием среды разработки Android Studio и облачной базы данных Firebase Firestore.

Целью проекта являлось упрощение процессов покупки, продажи автомобилей, управления объявлениями и взаимодействия между пользователями в мобильной среде.

В процессе выполнения курсового проекта были сформированы требования к системе, спроектированы архитектура приложения и базы данных, разработан пользовательский интерфейс и реализован основной функционал, включая авторизацию, регистрацию, создание, поиск и фильтрацию объявлений, работу с избранным, создание записей в боржурнале, а также административные функции для управления контентом.

Для хранения данных применялась облачная СУБД Firebase Firestore, для интеграции карт и геолокации — Yandex MapKit. Для оформления документации использовался текстовый редактор Microsoft Word 2016. База данных размещалась в облаке, что позволило обеспечить доступ к данным в реальном времени.

В процессе разработки возникали ошибки, связанные с обработкой сетевых запросов, сериализацией данных и обработкой изображений. Ошибки устранялись через дополнительную проверку данных, отладку сетевых взаимодействий и оптимизацию механизмов загрузки изображений.

Результаты тестирования показали, что основной функционал приложения работает стабильно, корректно обрабатывает пользовательский ввод, обеспечивает выполнение бизнес-процессов, предусмотренных техническим заданием. Имеются некоторые незначительные возможности для дальнейшего улучшения интерфейса и расширения возможностей фильтрации.

В целом, мобильное приложение Drivee полностью соответствует заявленным требованиям, демонстрирует стабильную работу, удобство использования и готово к эксплуатации пользователями.

СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

1. Создание адаптивного интерфейса для Android-приложений.
URL: <https://developer.android.com/guide/topics/ui/declaring-layout> (Дата обращения: 01.04.2025)
2. Руководство по работе с Firebase Firestore в Android.
URL: <https://firebase.google.com/docs/firestore/quickstart> (Дата обращения: 01.04.2025)
3. Авторизация и регистрация через Firebase Authentication.
URL: <https://firebase.google.com/docs/auth/android/start> (Дата обращения: 02.04.2025)
4. Настройка восстановления пароля в Firebase.
URL: https://firebase.google.com/docs/auth/android/manage-users#send_a_password_reset_email (Дата обращения: 05.04.2025)
5. Добавление ViewPager2 и перелистывания фотографий в Android.
URL: <https://developer.android.com/guide/navigation/navigation-swipe-view-2> (Дата обращения: 05.04.2025)
6. Работа с RecyclerView: отображение списков в Android.
URL: <https://developer.android.com/guide/topics/ui/layout/recyclerview> (Дата обращения: 06.04.2025)
7. Добавление избранного с помощью Firebase.
URL: <https://stackoverflow.com/questions/48650898/add-to-favorites-in-firebase-android> (Дата обращения: 08.04.2025)
8. Создание фильтрации данных в приложении Android.
URL: <https://stackoverflow.com/questions/24090968/filtering-recyclerview-using-a-searchview> (Дата обращения: 09.04.2025)
9. Интеграция карт и геолокации с помощью Yandex MapKit.
URL: <https://yandex.ru/dev/maps/mobile-sdk/doc/android/quickstart/concepts/android-quickstart-docpage/> (Дата обращения: 09.04.2025)

10. Подключение и настройка Bottom Navigation View.
URL: <https://developer.android.com/guide/navigation/navigation-ui> (Дата обращения: 12.04.2025)
11. Как работать с изображениями в Firebase Storage и отображать их в приложении.
URL: <https://firebase.google.com/docs/storage/android/start> (Дата обращения: 01.04.2025)
12. Права доступа в Android (запрос на доступ к звонкам, памяти и интернету).
URL: <https://developer.android.com/training/permissions/requesting> (Дата обращения: 15.04.2025)
13. Отправка звонка с приложения Android.
URL: <https://stackoverflow.com/questions/4275679/how-to-make-a-phone-call-in-android> (Дата обращения: 01.04.2025)
14. Создание форм регистрации с проверкой введённых данных.
URL: <https://developer.android.com/guide/topics/ui/form-validation> (Дата обращения: 15.04.2025)
15. Работа с фрагментами в Android и их навигацией.
URL: <https://developer.android.com/guide/fragments> (Дата обращения: 15.04.2025)
16. Создание custom-адаптеров для RecyclerView.
URL: <https://developer.android.com/guide/topics/ui/layout/recyclerview#java> (Дата обращения: 15.04.2025)
17. Основы архитектуры MVVM в Android.
URL: <https://developer.android.com/jetpack/guide> (Дата обращения: 15.04.2025)
18. Работа с облачными функциями в Firebase для сложной логики.
URL: <https://firebase.google.com/docs/functions> (Дата обращения: 15.04.2025)
19. Интеграция платежных систем в Android-приложения.
URL: <https://developer.android.com/google/play/billing> (Дата обращения: 16.04.2025)

20.Рекомендации по производительности мобильных приложений.

URL: <https://developer.android.com/topic/performance> (Дата обращения:
16.04.2025)