

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Разработка приложений для мобильных платформ»
Тема: Разработка приложения автоконтроль

Студент гр. 1303	_____	Иванов А.С.
Студент гр. 1303	_____	Насонов Я.К.
Студент гр. 1303	_____	Попандопуло А.Г.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург,
2025

ЗАДАНИЕ

Студент Иванов А.С.

Студент Насонов Я.К.

Студент Попандопуло А.Г.

Группа 1303

Тема: Разработка приложения автоконтроль

Исходные данные:

Необходимо разработать приложение, которое реализует сервис для контроля состояния автомобиля и помощи в его обслуживании. Пользователи — это автовладельцы (вводят данные автомобиля, получают напоминания, ведут журнал работ) и система (генерирует уведомления, сохраняет записи, предоставляет поиск).

Содержание пояснительной записки:

«Содержание»

«Введение»

«Сценарии использования»

«Пользовательский интерфейс»

«Разработанное приложение»

«Выводы»

«Список литературы»

«Приложение»

Предполагаемый объем пояснительной записки:
Не менее 15 страниц.

Дата выдачи задания: 11.02.2025

Дата сдачи реферата: 24.03.2025

Дата защиты реферата: 24.03.2025

Студент	_____	Иванов А.С.
Студент	_____	Насонов Я.К.
Студент	_____	Попандопуло А.Г.
Преподаватель	_____	Заславский М.М.

АННОТАЦИЯ

В рамках проекта было разработано мобильное приложение "АвтоКонтроль" для смартфонов с операционной системой Android. Приложение позволяет пользователям вводить данные об автомобиле, получать автоматические, а также пользовательские напоминания о техническом обслуживании, вести журнал выполненных работ, может осуществлять поиск и фильтрацию записей. Для хранения данных используется локальная база данных Room. Исходный код и дополнительную информацию можно найти в репозитории проекта: <https://github.com/moevm/adfmp1h25-auto>.

SUMMARY

As part of the project, a mobile application "AutoControl" was developed for Android smartphones. The application enables users to input vehicle data, receive automatic and custom maintenance reminders, maintain a service log, and search and filter records. The local Room database is used for data storage. The source code and additional information can be found in the project repository: <https://github.com/moevm/adfmp1h25-auto>.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	7
1.1. Актуальность проблемы.....	7
1.2. Постановка задачи.....	7
1.3. Предлагаемое решение.....	7
1.4. Обоснование реализации в виде мобильного приложения.....	8
2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ.....	9
1. Сценарий использования: Первичный ввод данных об автомобиле.....	9
2. Сценарий использования: Автоматическое создание напоминаний.....	10
3. Сценарий использования: Добавление напоминаний.....	10
4. Сценарий использования: Просмотр списка напоминаний.....	11
5. Сценарий использования: Поиск напоминаний и записей обслуживания	12
6. Сценарий использования: Добавление записи в журнал обслуживания	13
7. Сценарий использования: Просмотр журнала обслуживания.....	13
8. Сценарий использования: Просмотр экрана About	14
3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС.....	15
3.1. Макет интерфейса с графом переходов.....	15
3.2. Целевые устройства	16
4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ	17
4.1. Краткое описание	17
4.2. Используемые технологии	17
4.3. Используемые модули и системные библиотеки	17
5. ВЫВОДЫ	18
5.1. Достигнутые результаты.....	18
5.2. Недостатки и пути для улучшения полученного решения	18
5.3. Будущее развитие решения.....	18
6. СПИСОК ЛИТЕРАТУРЫ.....	19
7. ПРИЛОЖЕНИЯ	20

7.1. Инструкция для пользователя.....	20
7.2. Снимки экрана приложения.....	20

1. ВВЕДЕНИЕ

1.1. Актуальность проблемы

В современном мире автомобили являются неотъемлемой частью жизни миллионов людей. Однако поддержание автомобиля в исправном состоянии требует регулярного технического обслуживания, о котором пользователи часто забывают из-за плотного графика или недостатка информации. Это может привести или к поломкам, или увеличению затрат на ремонт, или же снижению безопасности на дорогах. Существующие решения, такие как бумажные журналы или сторонние приложения, часто неудобны, не персонализированы или требуют значительных усилий для ведения записей, например обычное использование блокнота.

1.2. Постановка задачи

Необходимо разработать мобильное приложение "АвтоКонтроль", которое поможет пользователям контролировать состояние своего автомобиля, получать персонализированные напоминания о техническом обслуживании и вести журнал выполненных работ. Приложение должно быть интуитивно понятным, удобным и доступным на мобильных устройствах.

1.3. Предлагаемое решение

Разработать мобильное приложение "АвтоКонтроль" для Android, которое дать пользователям возможность вводить данные об автомобиле, такие как марка, модель, год выпуска и пробег, получать автоматические напоминания о техобслуживании на основе этих данных и шаблонов, добавлять свои напоминания, вести журнал обслуживания с записями о работах, просматривать, фильтровать, а также сортировать записи, искать их по ключевым словам, в дополнение узнавать информацию о разработчиках.

1.4. Обоснование реализации в виде мобильного приложения

Мобильное приложение выбрано как достаточно оптимальный вариант: поскольку смартфоны всегда под рукой у пользователей, они отлично подходят для решения повседневных задач. Смартфоны также имеют возможность отправки push-уведомлений для своевременных напоминаний, позволяют удобно вести журнал сразу после выполнения работ и обеспечивают быстрый доступ к данным об автомобиле.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

Для определения требований к приложению был написан ряд предполагаемых сценариев использования:

1. Сценарий использования: Первичный ввод данных об автомобиле

Описание:

Пользователь вводит базовую информацию о своём автомобиле (марка, модель, год выпуска, начальный пробег, средне-ежедневный пробег), чтобы приложение могло генерировать персонализированные напоминания.

Действующие лица:

Пользователь, система.

Основной сценарий:

1. Пользователь переходит в раздел "Мой автомобиль" с главной страницы.
2. Система показывает форму с полями: "Марка", "Модель", "Год выпуска", "Текущий пробег" и др.
3. Пользователь заполняет поля и нажимает "Сохранить".
4. Система сопоставляет введенные данные и сохраняет их локально.
5. Система использует полученные данные для генерации персонализированных напоминаний (например, "Замена масла через 5 000 км").

Альтернативные сценарии:

- За. Если пользователь пропускает ввод данных:
 - За.1. Система показывает подсказку: "Добавьте данные о вашем автомобиле для персонализированных напоминаний".
 - За.2. Пользователь может оставить поля пустыми, но тогда некоторые функции (например, автоматические интервалы обслуживания) будут недоступны.

Результат:

Пользователь успешно ввёл данные об автомобиле, и система готова генерировать персонализированные напоминания.

2. Сценарий использования: Автоматическое создание напоминаний

Описание:

Система автоматически генерирует напоминания на основе заданного расписания и анализа данных о состоянии автомобиля.

Действующие лица:

Система (косвенно для пользователя).

Основной сценарий:

1. Система сопоставляет введенные об автомобиле данные с предустановленными шаблонами.
2. Система создаёт напоминания для проведения необходимых проверок и технического обслуживания.
3. Пользователь видит созданные напоминания в списке.

Альтернативные сценарии:

- 1а. Если система не может сопоставить данные (марка/модель) с предустановленными шаблонами:
 - 1а.1. Пользователь будет получать уведомления по среднестатистическим рекомендациям.

Результат:

Пользователь получает автоматически сгенерированные напоминания, основанные на данных автомобиля.

3. Сценарий использования: Добавление напоминаний

Описание:

Пользователь может самостоятельно добавить напоминание для контроля состояния автомобиля.

Действующие лица:

Пользователь, система.

Основной сценарий:

1. Пользователь переходит на главный экран и выбирает опцию "Добавить напоминание".

2. Система отображает всплывающее окно ввода с полями "Что напомнить?" (например, "Замена масла"), "Когда?" (календарь или выбор через ползунок), "Пробег" (опционально).
3. Пользователь заполняет поля (например, выбирает "Замена масла", дату через календарь и вводит "55 000 км" как целевой пробег) и нажимает "Сохранить".
4. Система добавляет напоминание в список и показывает его на главном экране или в разделе "Напоминания".

Альтернативные сценарии:

- За. Если пользователь оставляет поля пустыми:
За.1. Система показывает подсказку: "Пожалуйста, укажите, что нужно сделать".

Результат:

Пользователь добавил собственное напоминание, которое отображается в списке.

4. Сценарий использования: Просмотр списка напоминаний

Описание:

Пользователь может просматривать список напоминаний и фильтровать/сортировать их по дате или пробегу.

Действующие лица:

Пользователь, система.

Основной сценарий:

1. Пользователь переходит в раздел "Напоминания", где система отображает список всех напоминаний в хронологическом порядке.
2. Пользователь видит напоминания с цветовым выделением каждого из них. Наиболее близким по дате соответствует ярко-красный оттенок.
3. Пользователь нажимает на значок фильтрации и выбирает признак фильтрации, например "По дате", "По пробегу".
4. Пользователь нажимает кнопку "Применить".

5. Система обновляет список, показывая только отфильтрованные напоминания.

Альтернативные сценарии:

- 3а. Пользователь нажимает на значок сортировки и выбирает признак (дата):

3а.1. Система отображает отсортированный список напоминаний.

Результат:

Пользователь просматривает отфильтрованный или отсортированный список напоминаний.

5. Сценарий использования: Поиск напоминаний и записей обслуживания

Описание:

Пользователь может быстро найти конкретную запись в журнале обслуживания или напоминание по ключевым словам

Действующие лица:

Пользователь.

Основной сценарий:

1. Пользователь нажимает на иконку поиска (лупа) в верхней части одного из экранов ("Напоминания", "Журнал обслуживания").
2. Система открывает поле ввода с подсказкой: "Введите ключевое слово (например, 'масло', 'шины')".
3. Пользователь вводит сообщение для поиска.
4. Система отображает все напоминания, содержащие введенный текст.
5. Пользователь видит искомую запись.

Альтернативные сценарии:

- 4а. Если введенное сообщение не встречается в списке:
4а.1. Система уведомляет пользователя об отсутствии записи.

Результат:

Пользователь нашёл нужное напоминание или запись обслуживания.

6. Сценарий использования: Добавление записи в журнал обслуживания

Описание:

Пользователь фиксирует данные о проведённом техническом обслуживании, используя предустановленные шаблоны.

Действующие лица:

Пользователь, система.

Основной сценарий:

1. Пользователь переходит в раздел "Журнал обслуживания" и нажимает "Добавить запись".
2. Система показывает список шаблонов (например, "Замена масла", "Проверка шин", "ТО") или позволяет ввести собственное описание.
3. Пользователь выбирает один из шаблонов, и система автоматически заполняет поля: "Тип обслуживания" (в соответствии с шаблоном), "Дата" (текущая дата, но редактируемая), "Пробег", "Стоимость" (опционально).
4. Опционально пользователь может ввести подробное описание.
5. Система сохраняет запись локально и добавляет её в список журнала.

Альтернативные сценарии:

- За. Если пользователь хочет создать свою запись без шаблона:
За.1. Он выбирает "Другое" и заполняет все поля вручную.

Результат:

Пользователь добавил запись о проведённом обслуживании в журнал.

7. Сценарий использования: Просмотр журнала обслуживания

Описание:

Пользователь может просматривать журнал обслуживания и фильтровать/сортировать записи.

Действующие лица:

Пользователь, система.

Основной сценарий:

1. Пользователь переходит в раздел "Журнал обслуживания", где система отображает список всех записей в хронологическом порядке.
2. Пользователь нажимает на значок фильтрации и выбирает признак фильтрации, например "По дате", "По стоимости".
3. Пользователь нажимает кнопку "Применить".
4. Система обновляет список, отображая только отфильтрованные записи.

Альтернативные сценарии:

- 2а. Пользователь нажимает на значок сортировки и выбирает признак (дата или пробег):
2а.1. Система отображает отсортированный список записей обслуживания.

Результат:

Пользователь просматривает отфильтрованный или отсортированный журнал обслуживания.

8. Сценарий использования: Просмотр экрана About

Описание:

Пользователь может посмотреть страницу "Авторы".

Действующие лица:

Пользователь.

Основной сценарий:

1. Пользователь переходит на главную страницу.
2. Пользователь нажимает на иконку "About" (i).
3. Система отображает авторов приложения.

Результат:

Пользователь просмотрел информацию об авторах приложения.

3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

3.1. Макет интерфейса с графом переходов

На основании сценариев использования был разработан макет пользовательского интерфейса с графом переходов, представленный на рис. 1.

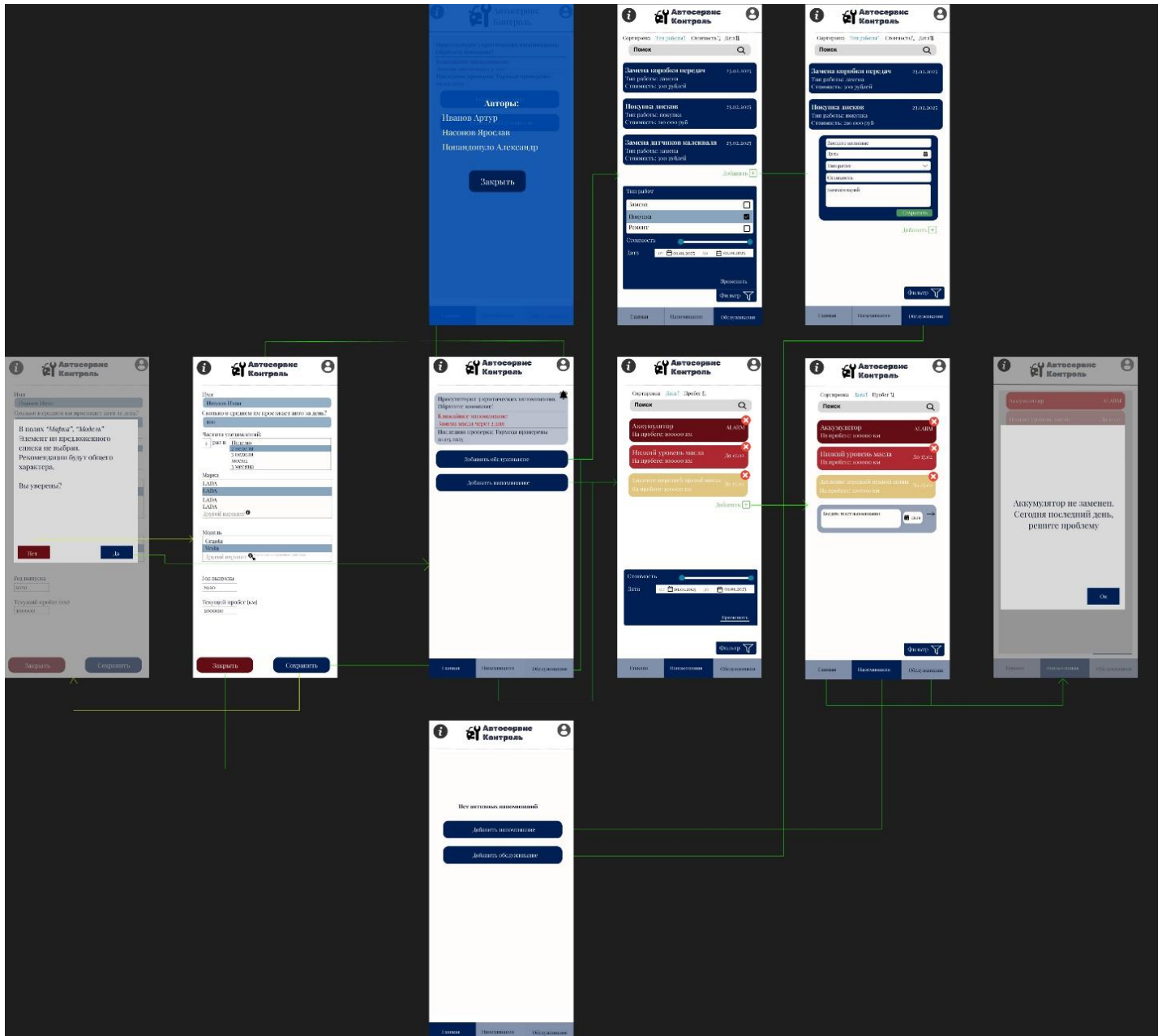


Рисунок 1 - Макет интерфейса с графом переходов.

3.2. Целевые устройства

Целевые устройства — смартфоны с ОС Android, так как:

- Android занимает значительную долю рынка.
- Обеспечивается легкая интеграция с сервисами Google.
- Широкий выбор устройств доступен пользователям.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

4.1. Краткое описание

Приложение "АвтоКонтроль" позволяет пользователям контролировать состояние автомобиля, получать напоминания о техническом обслуживании и вести журнал работ. Основной функционал включает ввод данных, автоматическую генерацию напоминаний, добавление записей и поиск.

4.2. Используемые технологии

- Jetpack Compose — для создания UI.
- Room — для локального хранения данных.
- Kotlin Coroutines — для асинхронных операций.
- Lifecycle-ViewModel — для управления данными и состоянием.

4.3. Используемые модули и системные библиотеки

Реализованное приложение использует основные компоненты Android, в частности Activity — для отображения UI и material-icons – как стандартный набор основных иконок

5. ВЫВОДЫ

5.1. Достигнутые результаты

Разработано приложение "АвтоКонтроль", которое автоматизирует контроль технического обслуживания автомобиля с помощью персонализированных напоминаний и удобного журнала.

5.2. Недостатки и пути для улучшения полученного решения

Отсутствие синхронизации между устройствами, решение – интеграция с облачным хранилищем. Также отсутствует автоматическое обновления пробега, решение – подключение к датчикам автомобиля (по возможности для современных автомобилей).

5.3. Будущее развитие решения

- Интеграция с сервисами для обновления пробега.
- Добавление напоминаний через email и мессенджеры (Telegram, VK)
- Разработка версии для iOS.

6. СПИСОК ЛИТЕРАТУРЫ

1. Репозиторий проекта // GitHub. URL: <https://github.com/moevm/adfmp1h25-auto>
2. Документация Android Studio // Developer.android.com. URL: <https://developer.android.com/guide>
3. Документация Jetpack Compose // Developer.android.com. URL: <https://developer.android.com/jetpack/compose>
4. Документация Kotlin // Kotlin.org. URL: <https://kotlinlang.org/docs/home.html>
5. Material Icons // Fonts.google.com URL: <https://fonts.google.com/icons>

4:51

Автосервис
Контроль

Сортировка **Дата** **Стоимость**

Поиск →

Обслуживание коробки передач
Тип работ: Замена
Дата: 05.03.2025 Стоимость: 10000 руб

Обслуживание дисков
Тип работ: Покупка
Дата: 06.03.2025 Стоимость: 8000 руб

Обслуживание датчиков
Тип работ: Замена
Дата: 07.03.2025 Стоимость: 3000 руб

Добавить +

Сбросить Фильтр

Главная Напоминания Обслуживания

Рисунок 4 – Экран обслуживания

4:51

Автосервис
Контроль

Имя

Введите имя

Сколько в среднем км проезжает авто в месяц?

Введите км 0

Марка автомобиля

Введите или выберите марку
Lada

Модель автомобиля

Введите или выберите модель
Vesta

Год выпуска

Выберите год
2025

Текущий пробег

Введите пробег (км)
0

Рекомендации по обслуживанию:
Замена масла: каждые 15000 км
Полная проверка: каждые 30000 км

Главная Напоминания Обслуживания

Рисунок 5 – Экран «Мой автомобиль»

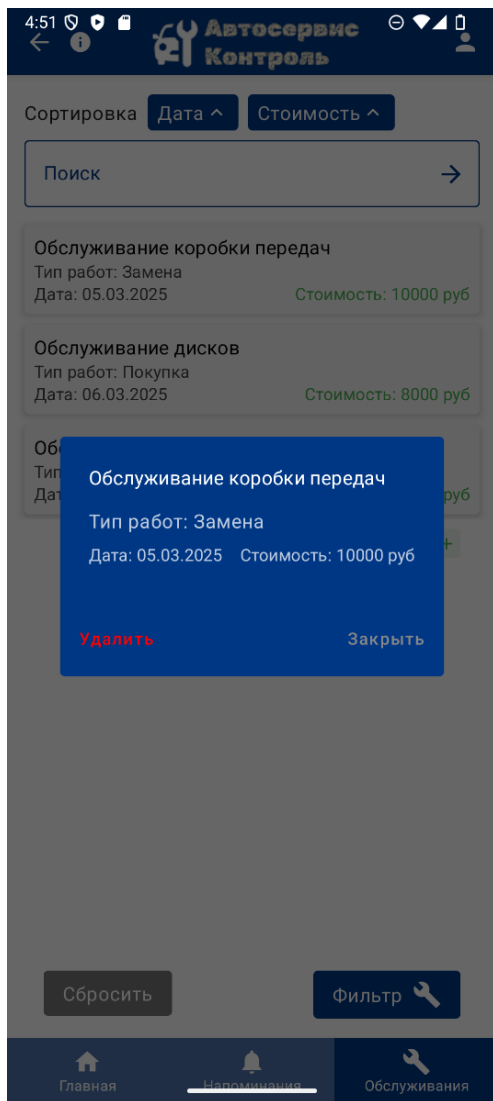


Рисунок 6 – Информация об обслуживании

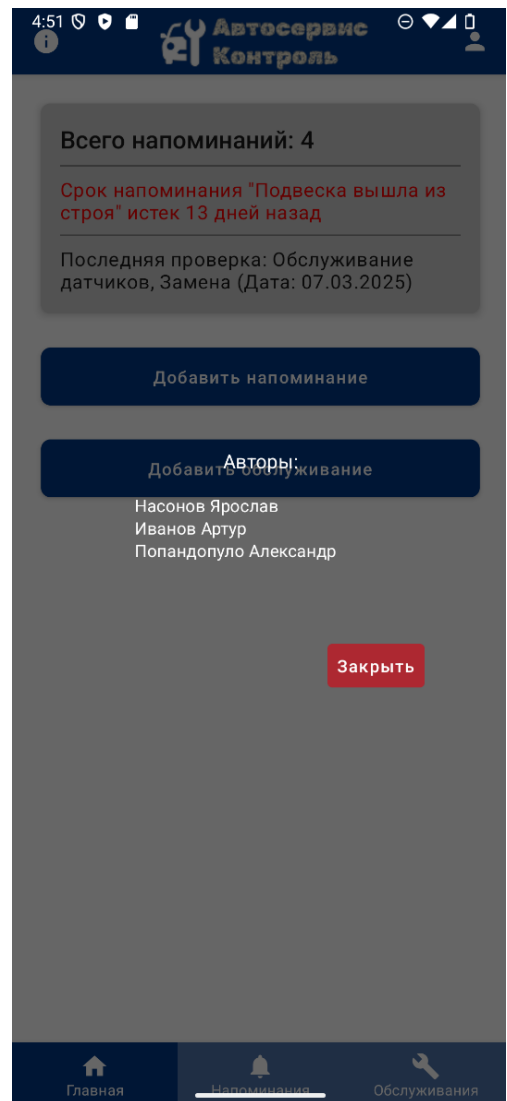


Рисунок 7 – Информация об авторах