

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

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Разработка приложений для мобильных платформ»**  
**Тема: Напоминатель для тревожных людей**

Студентка гр. 1303	_____	Королева П.А.
Студентка гр. 1303	_____	Куклина Ю.Н.
Студентка гр. 1303	_____	Сырцева Д.Д.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург  
2025

## ЗАДАНИЕ

Студентка Куклина Ю.Н.

Студентка Сырцева Д.Д.

Студентка Королева П.А.

Группа 1303

Тема: Напоминатель для тревожных людей

Исходные данные:

Необходимо реализовать андроид-приложение, реализующее напоминатель для тревожных и забывчивых людей.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Сценарии использования»

«Модель данных»

«Разработанное приложение»

«Выводы»

«Приложения»

«Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 05.02.2024

Дата сдачи реферата: 24.03.2024

Дата защиты реферата: 24.03.2024

Студентка	_____	Королева П.А.
Студентка	_____	Куклина Ю.Н.
Студентка	_____	Сырцева Д.Д.
Преподаватель	_____	Заславский М.М.

## АННОТАЦИЯ

В рамках курсовой работы разработано андроид-приложение, представляющее собой напоминатель для тревожных людей. Приложение включает функционал для создания дел, которые необходимо выполнить, их отметки, а также добавления фото и удаления. Реализована система фильтрации, сортировки и поиска задач по имени.

Для разработки использованы технологии Kotlin, Jetpack Compose.

Найти исходный код можно по ссылке: [reminder](#)

## SUMMARY

As part of the coursework, an Android application was developed, which is a reminder for anxious people. The application includes functionality for creating tasks that need to be completed, marking them, as well as adding photos and deleting them. A system for filtering, sorting and searching for tasks by name was implemented.

Kotlin and Jetpack Compose technologies were used for the development.

The source code can be found at the link: [reminder](#)

## СОДЕРЖАНИЕ

1.	Введение	7
1.1.	Актуальность проблемы	7
1.2.	Постановка задачи	7
1.3.	Предлагаемое решение	8
1.4.	Почему решение необходимо реализовать как мобильное приложение	8
2.	Сценарии использования	9
2.1.	Сценарий использования: Добавление нового дела	9
2.2.	Сценарий использования: Отметка выполненного дела	9
2.3.	Сценарий использования: Прикрепление фото	10
2.4.	Сценарий использования: Сброс всех отметок	11
2.5.	Сценарий использования: Сброс выполнения дела	11
2.6.	Сценарий использования: Поиск дела	12
2.7.	Сценарий использования: Фильтрация/Сортировка дел	12
2.8.	Сценарий использования: Просмотр страницы "О нас"	13
3.	Модель данных	14
3.1.	Макет интерфейса с графом переходов	14
3.2.	Целевые устройства, обоснование требований и максимально подробные характеристики	14
4.	Разработанное приложение	17
4.1.	Краткое описание	17
4.2.	Схема архитектуры	
4.2.1.	Клиентская часть	17
4.2.2.	Архитектура приложения	17
4.3.	Использованные технологии (внешние)	18
4.4.	Использованные модели/системные библиотеки платформы	18
6.	Выводы	20

6.1	Достигнутые результаты	20
6.2	Недостатки и пути для улучшения	20
6.3	Будущее развитие решения	20
7.	Литература	21
8.	Приложения	22
8.1	Инструкция для пользователя	22
8.2	Снимки экрана приложения	24

# **1. ВВЕДЕНИЕ**

## **1.1. Актуальность проблемы**

Тревожные расстройства являются одной из наиболее распространенных проблем психического здоровья в современном обществе. Люди, страдающие от повышенной тревожности, часто испытывают трудности с концентрацией, забывают о выполнении важных задач или постоянно сомневаются в том, выполнили ли они их. Это может приводить к стрессу, снижению качества жизни и даже к опасным ситуациям, например, если человек забыл выключить плиту или закрыть дверь. Существующие приложения-напоминатели часто не учитывают специфику тревожных людей, таких как необходимость подтверждения выполнения задачи (например, с помощью фото) или возможность быстрой фильтрации и сортировки задач. Поэтому создание специализированного приложения для этой аудитории является актуальной задачей.

## **1.2. Постановка задачи**

Задача проекта заключается в разработке андроид-приложения, которое позволит пользователям эффективно управлять своими повседневными задачами, снижая уровень тревожности и повышая уверенность в выполнении важных дел. Приложение должно предоставлять следующие возможности:

- Создание задач: пользователь может добавлять задачи с описанием, например, "выключить плиту" или "покормить кота".
- Подтверждение выполнения: возможность отмечать задачи как выполненные и добавлять фотографии в качестве доказательства выполнения (как из галереи, так и сделать снимок).
- Категоризация задач: разделение задач на категории для удобства организации (например, "дом", "работа", "личное").
- Поиск задач: быстрый поиск задач по названию.

- Сохранение данных: данные пользователя должны сохраняться между сессиями, чтобы при повторном входе в приложение все задачи оставались доступными.

Кроме того, приложение должно быть интуитивно понятным и минималистичным, чтобы не перегружать пользователя, что особенно важно для людей с тревожными расстройствами.

### **1.3. Предлагаемое решение**

Для реализации создается андроид-приложение с использованием Kotlin, Jetpack Compose. Данные сохраняются во внутреннем хранилище android устройства.

### **1.4. Почему решение необходимо реализовать как мобильное приложение**

- Мобильные устройства всегда находятся под рукой, что позволяет пользователю быстро добавлять задачи, проверять их статус и отмечать выполнение в любой момент. В отличие от компьютеров, которые не всегда доступны и не так удобны для оперативного использования в повседневной жизни, смартфоны обеспечивают мгновенный доступ к приложению в любой ситуации.

- Использование камеры: Мобильные устройства оснащены камерами, что позволяет пользователю добавлять фото в качестве подтверждения выполнения задачи.



## **2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ**

### **2.1. Сценарий использования: Добавление нового дела**

#### **Цель:**

Пользователь создает новую задачу в списке дел.

#### **Действующее лицо:**

Пользователь

#### **Основной сценарий:**

1. Пользователь открывает приложение и нажимает кнопку "Добавить дело".
2. Система открывает экран добавления нового дела.
3. Пользователь вводит название дела (например, "Заккрыть окно").
4. Пользователь выбирает категорию (например, "Бытовые дела").
5. Пользователь нажимает "Добавить".
6. Система сохраняет дело и добавляет его в общий список.

#### **Альтернативные сценарии:**

4а. Пользователь наживает "Добавить категорию" 4а.1 Система открывает окно с текстовым полем.

4а.2 Пользователь вводит название и нажимает "Сохранить".

4а.3 Система закрывает окно и указывает записанную категорию в поле "Категория".

#### **Результат:**

Дело успешно добавлено в список и появится в общем списке дел.

### **2.2. Сценарий использования: Отметка выполненного дела**

#### **Цель:**

Пользователь отмечает, что выполнил дело.

#### **Действующее лицо:**

Пользователь

#### **Основной сценарий:**

1. Пользователь открывает приложение и видит список дел.
2. Пользователь нажимает на чекбокс рядом с выполненным делом.

3. Система фиксирует выполнение дела и помечает его зеленым цветом.

**Альтернативные сценарии:**

3а. Ошибка при сохранении отметки.

3а.1 Система уведомляет пользователя и предлагает повторить попытку.

**Результат:**

Дело помечено как выполненное.

**2.3. Сценарий использования: Прикрепление фото**

**Цель:**

Пользователь прикрепляет фото выполненного дела.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Вызывается сценарий «Отметка выполненного дела»
2. Пользователь нажимает на иконку камеры рядом с делом.
3. Открывается всплывающее окно с выбором метода загрузки фото - из галереи или сделать снимок.
4. Пользователь выбирает "Сделать снимок".
5. Открывается камера.
6. Пользователь делает снимок.
7. Открывается превью фото с возможностью переснять или использовать фото.
8. Пользователь выбирает "Использовать фото".
9. Фото отображается В поле выбранного дела.

**Альтернативные сценарии:**

3а. Пользователь выбирает "Загрузить из галереи".

3а.1 Система открывает галерею в всплывающем окне.

3а.2 Пользователь выбирает фото.

3а.3 Переход к п.9 основного сценария.

8а. Пользователь выбирает переснять. 8а.1 Переход к п.6 основного сценария.

**Результат:**

К выполненному делу прикреплено подтверждающее фото.

**2.4. Сценарий использования: Сброс всех отметок**

**Цель:**

Пользователь очищает все выполненные дела.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь заходит на страницу со списком дел.
2. Пользователь нажимает кнопку "Сбросить все отметки".
3. Система запрашивает подтверждение "Вы уверены, что хотите удалить все отметки?".
4. Пользователь нажимает кнопку "Да".
5. Система удаляет все отметки, теперь все чекбоксы пустые.

**Альтернативные сценарии:**

4а. Пользователь нажимает кнопку "Отмена".

4а.1 Модальное окно закрывается, пользователь остается на странице со списком дел.

**Результат:**

Все отметки на делах очищены.

**2.5. Сценарий использования: Сброс выполнения дела**

**Цель:**

Пользователь отменяет выполнение одного дела.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь заходит на страницу со списком дел.
2. Пользователь на выбранный чекбокс рядом с делом, выполнение которого хочет отменить.
3. Система очищает чекбокс, он пустой и удаляет фото(если было).

**Результат:**

Выполнение дела отменено.

## **2.6. Сценарий использования: Поиск дела**

**Цель:**

Пользователь находит дело в списке.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение и нажимает на строку поиска.
2. Пользователь вводит ключевое слово.
3. Система отображает все дела, содержащие это слово.
4. Пользователь переходит на нужное ему дело.

**Альтернативные сценарии:**

4а. Введенное слово не найдено в списке дел. 4а.1 Система сообщает, что результатов нет, и предлагает создать новое дело.

**Результат:**

Пользователь нашел дело в списке.

## **2.7. Сценарий использования: Фильтрация/Сортировка дел**

**Цель:**

Пользователь фильтрует/сортирует дела по определенному признаку.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь переходит на страницу списка дел.

2. Пользователь нажимает на значок фильтрации.
3. Пользователь выбирает признак фильтрации (только отмеченные или только с фото).
4. Система отображает отфильтрованные дела из списка.

**Альтернативные сценарии:**

- 2а. Пользователь нажимает на значок сортировки.
- 2а.1 Система предлагает отсортировать дела по времени отметки.
- 2а.2 Пользователь нажимает кнопку "Применить".
- 2а.3 Система отображает дела в отсортированном порядке.

**Результат:**

Пользователь просматривает отфильтрованный/отсортированный список.

**2.8. Сценарий использования: Просмотр страницы "О нас"**

**Цель:**

Пользователь просматривает страницу "О нас".

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь попадает на главную страницу.
2. Пользователь выбирает в меню "О нас".
3. Система отображает страницу "О нас".

**Результат:**

Пользователь просматривает страницу about.

### 3. МОДЕЛЬ ДАННЫХ

#### 3.1. Макет интерфейса с графом переходов

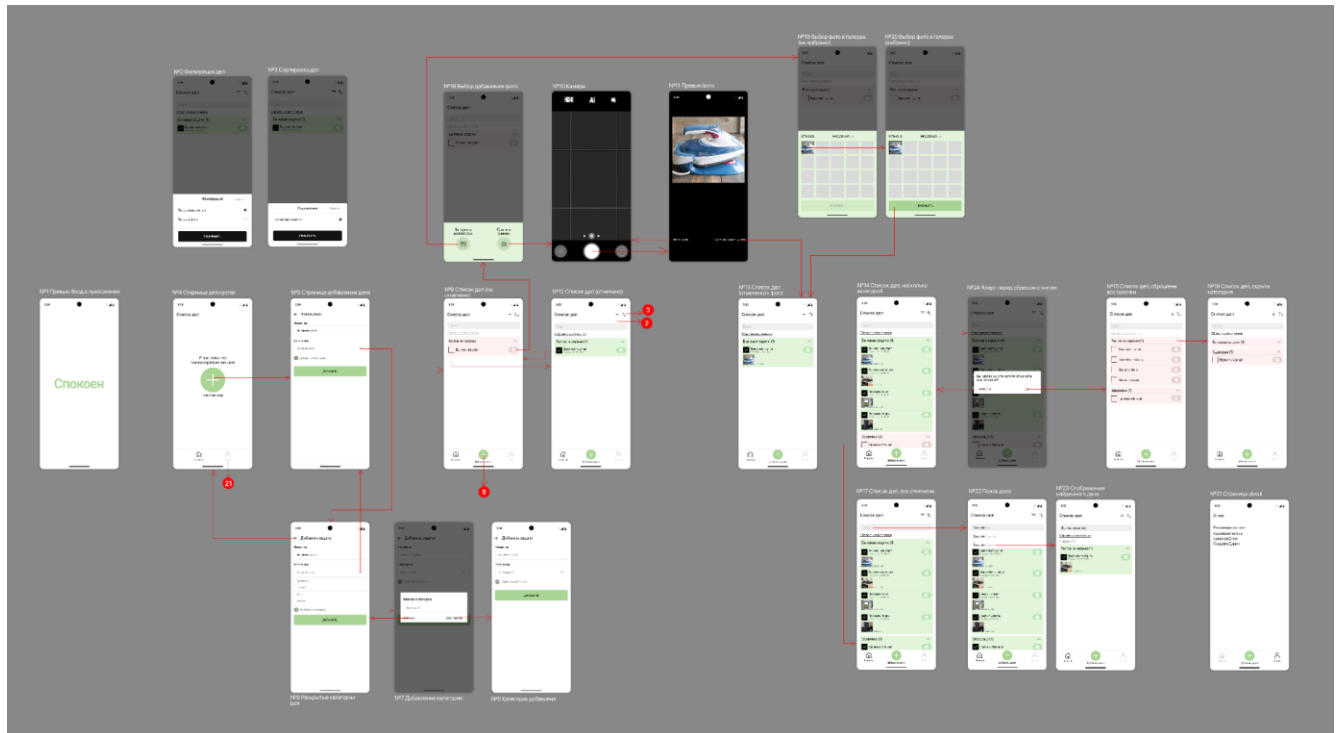


Рисунок 1. Макет интерфейса с графом переходов

#### 3.2 Целевые устройства, обоснование требований и максимально подробные характеристики

Тип устройства: Смартфон

Модель: Google Pixel 9 Pro

Характеристики:

Версия API 35

Экран:

hw.lcd.density 480

hw.lcd.height 2856

hw.lcd.width 1280

Остальное:

avd.ini.displayname Pixel 9 Pro API 35

avd.ini.encoding UTF-8

AvdId Pixel\_9\_Pro\_API\_35

disk.dataPartition.size 2G  
fastboot.chosenSnapshotFile  
fastboot.forceChosenSnapshotBoot no  
fastboot.forceColdBoot no  
fastboot.forceFastBoot yes  
hw.accelerometer yes  
hw.arc false  
hw.audioInput yes  
hw.battery yes  
hw.camera.back virtualscene  
hw.camera.front emulated  
hw.cpu.ncore 2  
hw.device.hash2 MD5:73e7b35d09e3a8055043aca4688e0dad  
hw.device.manufacturer Google  
hw.device.name pixel\_9\_pro  
hw.dPad no  
hw.gps yes  
hw.gpu.enabled yes  
hw.gpu.mode auto  
hw.initialOrientation portrait  
hw.keyboard yes  
hw.mainKeys no  
hw.ramSize 11548  
hw.sdCard yes  
hw.sensors.orientation yes  
hw.sensors.proximity yes  
hw.trackBall no  
image.androidVersion.api 35  
image.sysdir.1 system-images/android-35/google\_apis\_playstore/x86\_64/  
PlayStore.enabled true

runtime.network.latency none  
runtime.network.speed full  
showDeviceFrame yes  
skin.dynamic yes  
tag.display Google Play  
tag.displaynames Google Play  
tag.id google\_apis\_playstore  
tag.ids google\_apis\_playstore  
vm.heapSize 256



## **4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ**

### **4.1 Краткое описание**

Проект представляет собой мобильное приложение для создания и управления напоминаниями. Пользователи могут создавать категории дел, добавлять задачи, отмечать выполнение, прикреплять подтверждающие фото, а также сортировать и фильтровать дела по различным параметрам. Также есть список со всеми делами, разделенными по категориям. Благодаря разделению на категории навигация по списку дел выполняется очень быстро.

В каждом деле написано его название, время отметки о выполнении (если есть), фото, подтверждающее выполнение задачи (если есть), время, когда была сделана фотография (если изображение прикреплено), отображается также статус выполнения. Фотографию можно сделать на лету или взять из галереи. Помимо этого есть страница “О нас”.

Приложение фокусируется на удобной организации повседневных дел, визуальном контроле их выполнения. Все данные хранятся локально на устройстве, обеспечивая быстрый доступ и офлайн-работу.

### **4.2 Схема архитектуры**

#### **4.2.1 Клиентская часть**

- **Язык:** Kotlin
- **UI:** Jetpack Compose

#### **4.2.2 Архитектура приложения**

##### **Data (Хранилище данных)**

FileManager.kt: управляет сохранением и загрузкой файлов, информации о делах.

##### **UI (Пользовательский интерфейс)**

##### **Компоненты (components):**

- PhotoPickerSheet.kt: Компонент, предоставляющий выбор (загрузки из галереи или фото “налету”).

- `TabBar.kt`: Навигационное меню.
- `TaskCategory.kt`: Компонент категории.
- `TaskItem.kt`: Элемент задачи.

#### **Модели (models):**

- `Task.kt`: Класс, представляющий структуру элемента дела.

#### **Страницы (pages):**

- `AboutScreen.kt`: Экран "О нас".
- `AddTask.kt`: Экран добавления новой задачи.
- `CameraScreen.kt`: Экран для съемки фото.
- `Empty.kt`: Экран пустого списка задач.
- `MainScreen.kt`: Главный экран приложения, отображает список задач или заглушку при их отсутствии.
- `TaskListScreen.kt`: Экран списка задач.

#### **Темы (theme)**

- `Color.kt`, `Theme.kt`, `Type.kt`: Определяют стили и цветовую схему интерфейса.

#### **Главный модуль**

`MainActivity.kt`: Основной класс, управляющий запуском приложения и его жизненным циклом.

### **4.3 Используемые технологии (внешние)**

В приложении используются только встроенные технологии, без внешних сервисов и сторонних библиотек. Все данные хранятся локально, без использования облачных хранилищ.

### **4.4 Используемые модули/системные библиотеки платформы UI, Jetpack Compose**

- `androidx.compose.material3:material3`

Используется для Material 3 компонентов интерфейса.

- `androidx.compose.ui:ui`  
Базовый модуль Jetpack Compose.
- `androidx.compose.ui:ui.graphics`  
Работа с графикой и цветами.
- `androidx.compose.ui:ui.tooling.preview`  
Предпросмотр UI-компонентов в Android Studio.
- `androidx.activity:activity-compose`  
Интеграция Jetpack Compose с жизненным циклом Activity.

### **Навигация и жизненный цикл**

- `androidx.navigation:navigation-compose`  
Управление навигацией между экранами в Compose.
- `androidx.lifecycle:lifecycle-runtime-kt`  
Управление жизненным циклом компонентов.

### **Работа с данными, изображениями**

- `com.google.code.gson:gson`  
Сериализация и десериализация JSON.
- `io.coil-kt:coil-compose`  
Загрузка и отображение изображений.

### **Работа с камерой**

- `androidx.camera:camera-core`  
Основная библиотека для работы с камерой.
- `androidx.camera:camera-camera2`  
Поддержка API Camera2.
- `androidx.camera:camera-lifecycle`  
Управление жизненным циклом камеры.
- `androidx.camera:camera-view`  
Виджет для работы с камерой в Compose.

## **6. ВЫВОДЫ**

### **6.1 Достигнутые результаты**

В ходе работы было разработано мобильное приложение "Reminder", представляющее собой список дел, которое позволяет пользователям добавлять дела, отмечать их выполненными, следить за временем отметки, прикреплять фотографии для подтверждения выполнения, снимать отметки и удалять дела.

### **6.2 Недостатки и пути для улучшения**

- 1) На данный момент приложение не предоставляет возможности удаления категорий, что приводит к заполнению списка категорий не актуальными вариантами. Для решения этой проблемы необходимо продумать поведение приложения при удалении категории, к которой принадлежат существующие дела, и добавить функциональность удаления категории.
- 2) Приложение не спрашивает пользователя при удалении дела и снятии единичной отметки о выполнении. Для решения этой проблемы необходимо добавить всплывающие диалоговые окна для подтверждения действия.

### **6.3 Будущее развитие решения**

В качестве развития решения возможны:

- 1) Локализация для поддержки нескольких языков.
- 2) Перенос данных на сервер, для возможности просмотра отмеченных дел на разных устройствах.
- 3) Совместное ведение списка дел для сожителей, коллег и т.п.
- 4) Автоматическое снятие отметки по времени, для регулярных дел.
- 5) Добавление напоминаний о невыполненных делах.

## 7. ЛИТЕРАТУРА

1. Ссылка на GitHub. - [Электронный ресурс]. - URL: <https://github.com/moevm/adfmp1h25-remind> (дата обращения: 21.03.2025).
2. Android Studio. – [Электронный ресурс]. – URL: <https://developer.android.com/studio> (дата обращения: 21.03.2025).
3. Kotlin. – [Электронный ресурс]. – URL: <https://kotlinlang.org/> (дата обращения: 21.03.2025).
4. Jetpack Compose. – [Электронный ресурс]. – URL: <https://developer.android.com/jetpack/compose> (дата обращения: 21.03.2025).

## **8. ПРИЛОЖЕНИЯ**

### **8.1 Инструкция для пользователя.**

#### **Добавление нового дела.**

Для того, чтобы добавить новое дело, необходимо нажать на кнопку “Добавить дело” на стартовой странице или в нижнем меню. После этого осуществится переход на страницу с формой, в которой нужно указать название нового дела и выбрать категорию из списка. Если нужной категории нет, ее можно создать, нажав на кнопку “Добавить категорию”. В открывшемся окне нужно указать название новой категории.

Названия дела и категории не могут быть пустыми, о чем приложение заранее предупреждает пользователя под формой для ввода. В случае если пользователь неправильно заполнил поле, оно подсветится красным цветом.

#### **Отметка выполненного дела.**

Для того, чтобы отметить дело, как выполненное, достаточно нажать чекбокс. Дело подсветится зеленым, появится временная отметка, когда дело было отмечено. Опционально, пользователь может добавить фотографию подтверждения выполнения дела. Фотографию можно загрузить из галереи или сделать фото.

#### **Снятие отметки о выполнении.**

Для того, чтобы снять отметку о выполнении, пользователь может нажать чекбокс на выполненном деле. Приложение удостоверится, действительно ли пользователь хочет снять отметку. После этого будут удалены фото и временная отметка, дело подсветится красным.

#### **Удаление дела.**

Для того, чтобы удалить дело, нужно смахнуть его влево в списке всех дел. При протягивании дела влево, появится иконка удаления, и, если продолжить, дело удалится.

### **Просмотр дел.**

Пользователь может просматривать дела на главной странице. В верхней части экрана он может воспользоваться регистронезависимым поиском по названию или категории, сортировкой по времени отметки:

- 1) по убыванию
- 2) по возрастанию

и фильтрацией:

- 1) только с фото
- 2) только отмеченные
- 3) только не отмеченные.

## 8.2 Снимки экрана приложения.

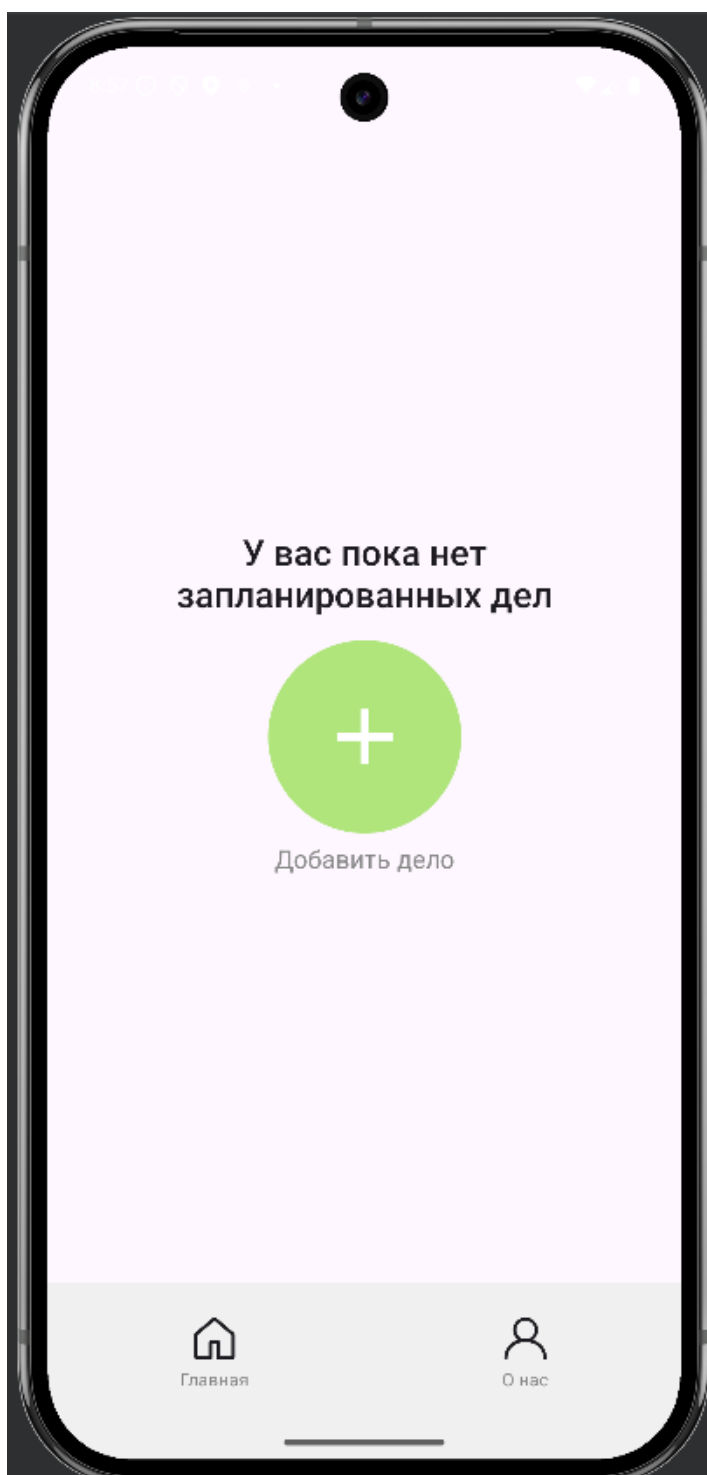


Рисунок 8.1 - Пустая главная страница





Рисунок 8.2 - Страница “О нас”

The image shows a mobile application interface for adding a task. At the top, there is a back arrow and the title "Добавить задачу". Below this is a form with two main sections. The first section is labeled "Название" and contains a text input field. Below the input field is a small error message: "Поле не может быть пустым". The second section is labeled "Категория" and contains a dropdown menu with the selected value "Быт" and a downward arrow. Below the dropdown is a link with a plus icon and the text "Добавить категорию". At the bottom of the form is a large green button with the text "ДОБАВИТЬ". The entire interface is displayed within a black frame representing a smartphone.

← Добавить задачу

Название

Поле не может быть пустым

Категория

Быт

+ Добавить категорию

ДОБАВИТЬ

Рисунок 8.3 - Страница создания дела

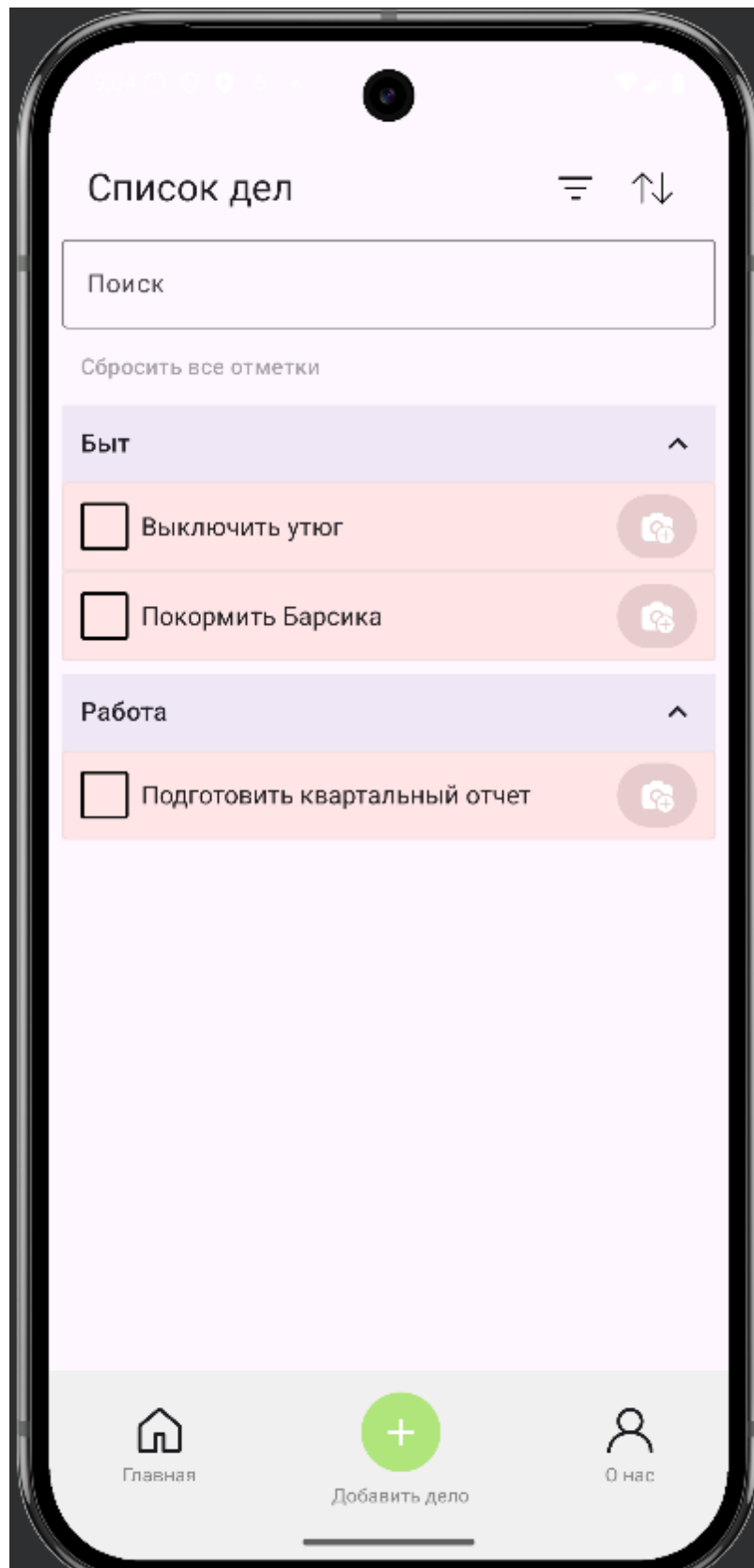


Рисунок 8.4 - Непустая главная страница

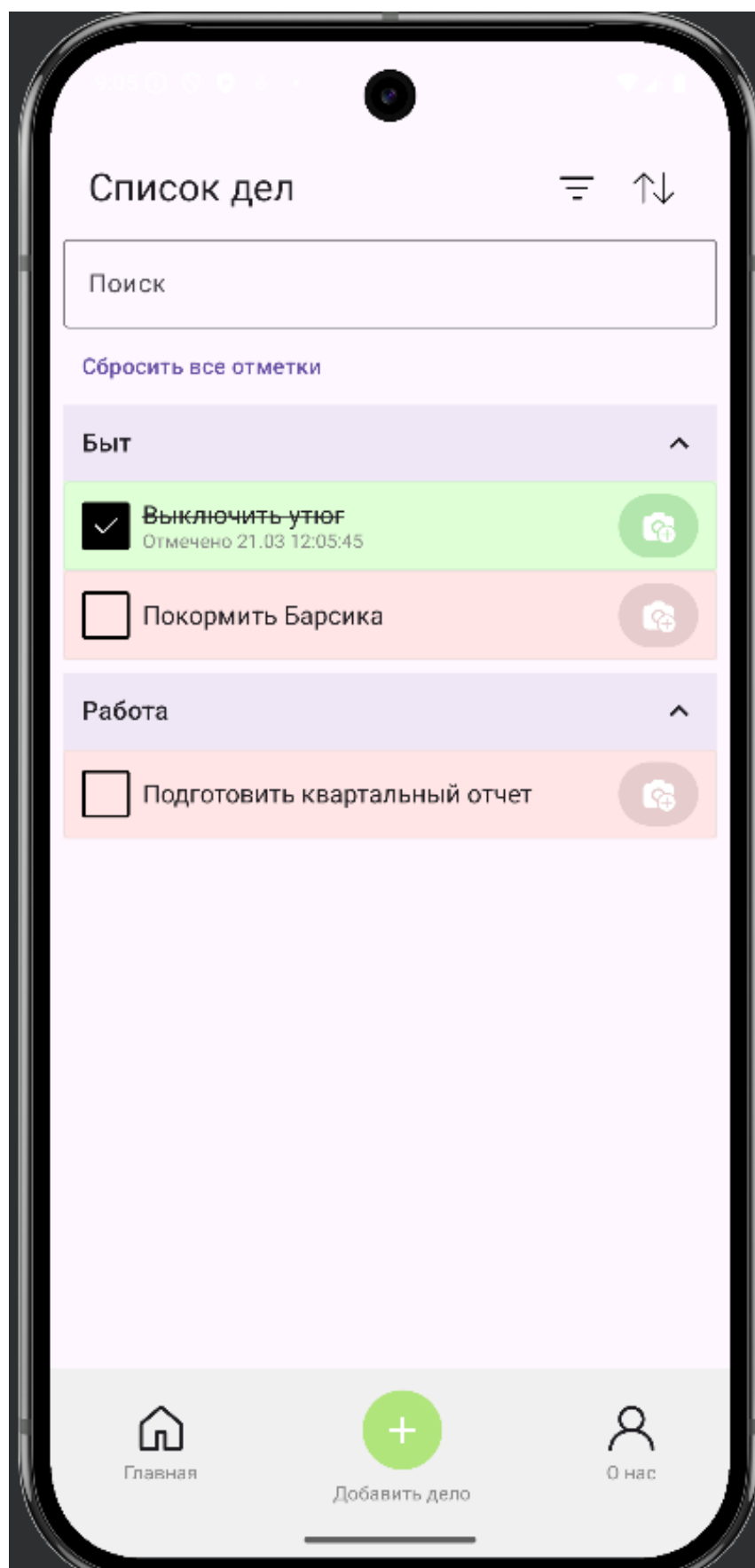


Рисунок 8.5 - Отмеченное дело

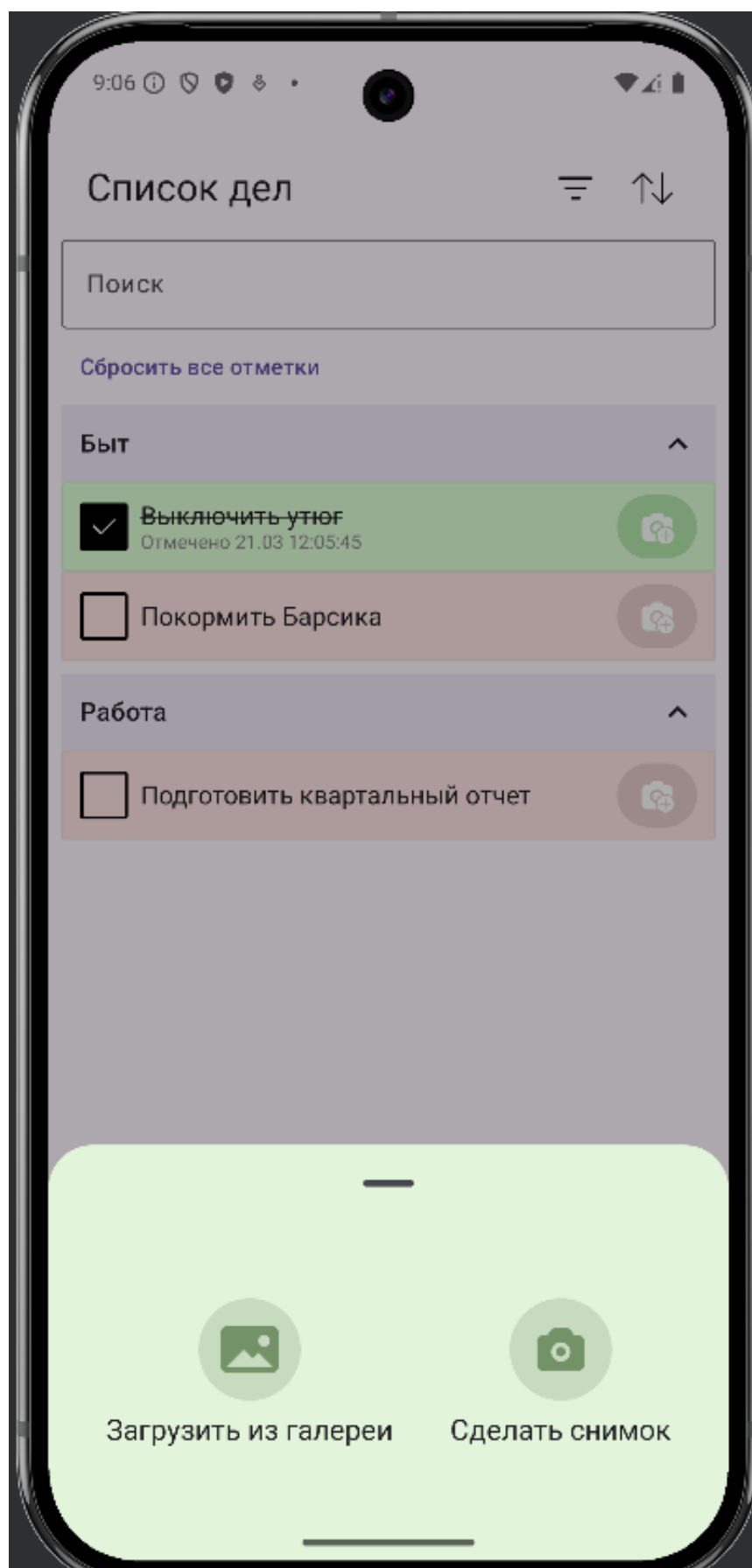


Рисунок 8.6 - Выбор фотографии для подтверждения



Рисунок 8.7 - Фотографирование в приложении

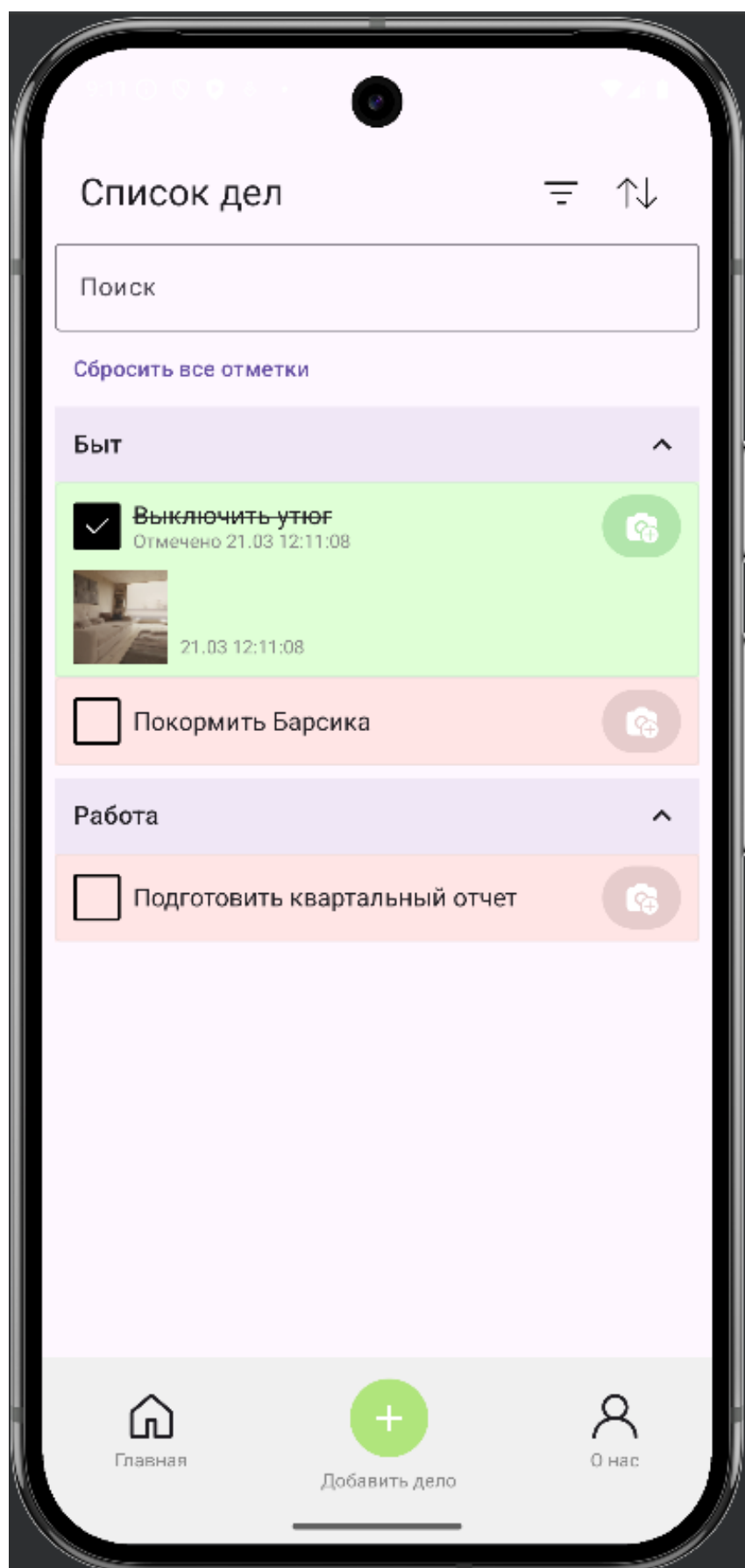


Рисунок 8.8 - Отмеченное дело с фотографией

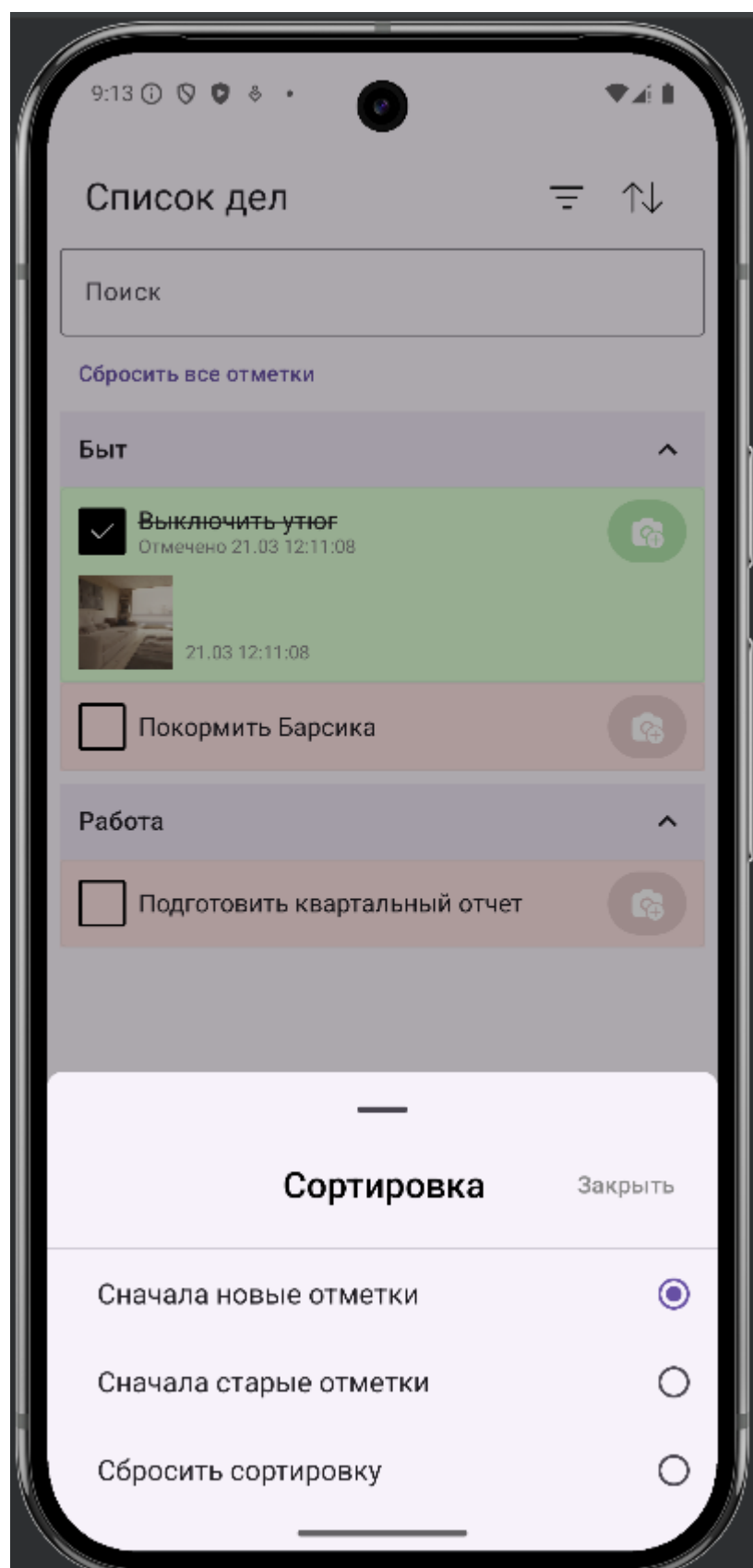


Рисунок 8.9 - Варианты сортировки



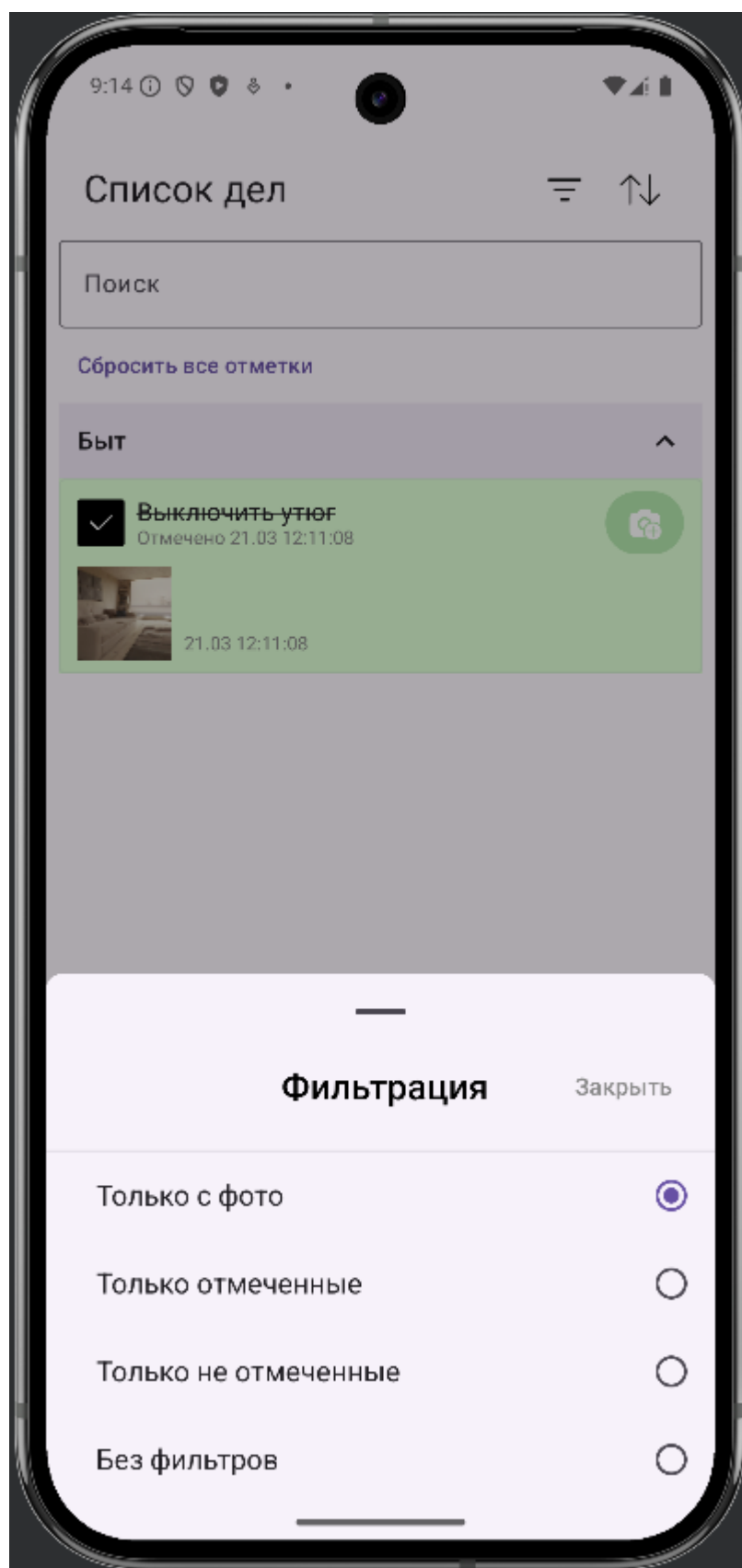


Рисунок 8.10 - Варианты фильтрации