МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пензенский государственный университет»**

|  |  |  |
| --- | --- | --- |
| *Факультет* |  | *Кафедра* |
| вычислительной техники |  | «Вычислительная техника» |

|  |  |
| --- | --- |
| Направление подготовки: | 09.03.01 «Информатика и вычислительная техника» |
| Профиль подготовки: | «Программное обеспечение средств вычислительной техники и автоматизированных систем» |

**БАКАЛАВРСКАЯ РАБОТА**

на тему

**АНДРОИД-ПРИЛОЖЕНИЕ**

**«ПЛАНИРОВЩИК ЗАДАЧ»**

**Студент** \_\_\_\_\_\_\_\_\_\_\_*31.05.2023* Д.О. Гусев

(подпись, дата) (ФИО полностью)

**Руководитель** \_\_\_\_\_\_\_\_\_\_\_*31.05.2023\_*  О.В. Юрова

(подпись, дата) (фамилия, инициалы)

**Нормоконтролёр** \_\_\_\_\_\_\_\_\_\_\_*31.05.2023* А.С. Бычков

(подпись, дата) (фамилия, инициалы)

*Работа допущена к защите* (протокол зас. кафедры от *02.06.23* г. № \_\_\_14\_\_\_\_)

**Заведующий кафедрой** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Митрохин

(подпись) (фамилия, инициалы)

*Работа защищена с отметкой* \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (протокол заседания ГЭК от

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_№\_\_\_\_\_\_\_\_\_)

**Секретарь ГЭК** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.И. Гурин

(подпись) (фамилия, инициалы)

Пенза 2023

**Реферат**

Бакалаврская работа содержит 92 листа, 34 рисунка, 1 таблицу, 5 источников и 3 приложения.

ANDROID, ANDROID STUDIO, JAVA, ROOM, *ПЛАНИРОВЩИК ЗАДАЧ*, *TODOLIST*.

Цель работы – разработать Андроид-приложение.

Объектом разработки является приложение Планировщик задач на операционной системе Android.

В результате проделанной работы было разработано приложение для добавления, удаления и напоминания задач. Проведено тестирование и отладка каждого элемента приложения.

**Содержание**

[**Перечень сокращений и обозначений** 5](#_Toc137747582)

[**1 Андроид-приложение «Планировщик задач»** 7](#_Toc137747583)

[**1.1 Анализ предметной области** 7](#_Toc137747584)

[**1.1.1 Анализ существующих аналогов** 7](#_Toc137747585)

[**1.1.2 Постановка задачи** 18](#_Toc137747586)

[**1.1.3 Выбор инструментов разработки** 19](#_Toc137747587)

[**1.1.4 Выбор языка программирования** 22](#_Toc137747588)

[**1.1.5 Выбор решения для хранения данных** 23](#_Toc137747589)

[**1.1.6 Выводы** 25](#_Toc137747590)

[**1.2 Реализация приложения** 25](#_Toc137747591)

[**1.2.1 layouts** 25](#_Toc137747592)

[**1.2.2 java-классы для работы с БД** 27](#_Toc137747593)

[**1.2.3 java-классы для работы с активити** 27](#_Toc137747594)

[**1.2.4 Выводы** 28](#_Toc137747595)

[**1.3 Тестирование приложения** 28](#_Toc137747596)

[**1.3.1 Тестирование** 28](#_Toc137747597)

[**1.3.2 Выводы** 34](#_Toc137747598)

[**2 Безопасность жизнедеятельности** 35](#_Toc137747599)

[**2.1 Анализ негативных факторов** 35](#_Toc137747600)

[**2.1.1 Оборудование рабочего места** 35](#_Toc137747601)

[**2.1.2 Освещенность рабочего места** 37](#_Toc137747602)

[**2.1.3 Электробезопасность** 37](#_Toc137747603)

[**2.1.4 Пожарная безопасность** 38](#_Toc137747604)

[**2.1.5 Режим труда и отдыха** 38](#_Toc137747605)

[**Заключение** 40](#_Toc137747606)

[**Список использованных источников** 41](#_Toc137747607)

[**Приложение А** 42](#_Toc137747608)

[**Приложение Б** 46](#_Toc137747609)

[**Приложение В** 62](#_Toc137747610)

## **Перечень сокращений и обозначений**

В настоящей работе применяют следующие сокращения и обозначения

ГОСТ – региональный стандарт, принятый Межгосударственным советом по стандартизации, метрологии и сертификации Содружества Независимых Государств

ООП – объектно-ориентированное программирование

ПК – персональный компьютер

БД – база данных

ОС - операционная система

ЭВМ – электронная вычислительная машина

*API* – ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Application Programming Interface*) – интерфейс прикладного программирования

*IDE* - ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Integrated Development Environment*) - единая среда разработки

*SQL* - (англ. *Structured Query Language*) - язык структурированных запросов

*UI* - ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *User Interface*) – пользовательский интерфейс

*USB* - ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Universal Serial Bus*) — универсальная последовательная шина

**Введение**

В настоящее время эффективное планирование задач – неотъемлемая часть жизни любого человека, и особенно актуально это в контексте работы. Постоянно возникает необходимость управления временем и ресурсами, планирования проектов, распределения задач и назначения сроков для их выполнения. Именно здесь на помощь приходит такой инструмент как планировщик задач.

Планировщик задач – это программное обеспечение, которое предоставляет возможность планировать, отслеживать и контролировать процесс выполнения задач. С его помощью удобно распределять задачи между сотрудниками, устанавливать приоритеты, определять сроки и отслеживать выполнение задачи. Кроме того, планировщик задач сокращает время на поиск информации, увеличивает производительность работы коллектива, а также обеспечивает точность выполнения задач и своевременное извещение о скором их истечении.

Целью данной работы является анализ существующих планировщиков задач, оценка их функциональных возможностей и разработка своего собственного проекта на операционной системе *Android*. В работе будет рассмотрено понятие планировщика задач, его основные функции и возможности, принципы построения таких систем. Также будут проанализированы существующие решения, проведено исследование их преимуществ и недостатков. В итоге будет разработан собственный планировщик задач с учетом всех вышеперечисленных факторов.

## **1 Андроид-приложение «Планировщик задач»**

## **1.1 Анализ предметной области**

В настоящее время люди сталкиваются с высокой загруженностью и растущим объемом работы и задач, которые нужно выполнить. В этом контексте приложение планировщик задач на андроид является важным инструментом для безопасного и эффективного управления задачами и временем. Оно обеспечивает возможность создавать и отслеживать задачи, установить приоритет и дедлайны, а также организовать график работы.

Приложения для планирования задач на *Android* доступны в различных вариантах, включая бесплатные и платные, легкий и полнофункциональный, и подходят для всех типов пользователей. Они могут быть полезны и в частных, и в профессиональных целях, так как справляются со сложной и многоуровневой организацией работы.

Таким образом, приложение планировщик задач на *Android* актуально и востребовано в нашем быстром и сильно загруженном обществе, и является незаменимым инструментом для тех, кто стремится улучшить свою работоспособность и эффективность.

## **1.1.1 Анализ существующих аналогов**

Среди аналогов можно выделить:

- «*Google* задачи»;

- «*Todolist*: планы и задачи»;

- «*TickTick*: список дел и задачи»;

- «*Microsoft To Do: Lists & Tasks*».

«*Google* задачи»

Пользовательский графический интерфейс представлен на рисунках 1 - 4.

Ссылка на скачивание:

[*https://play.google.com/store/apps/details?id=com.google.android.apps.tasks&hl=ru&gl=US*](https://play.google.com/store/apps/details?id=com.google.android.apps.tasks&hl=ru&gl=US)



Рисунок 1 – Логотип приложения в *Google Play*

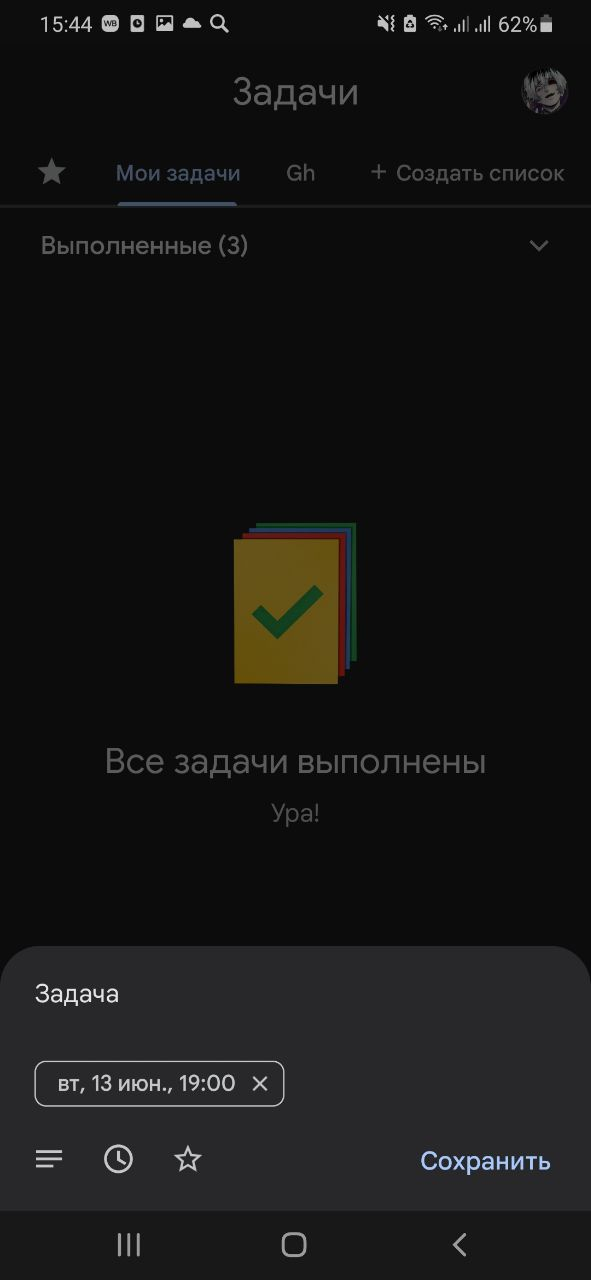


Рисунок 2 – Создание задачи



Рисунок 3 – Создание списка



Рисунок 4 – Созданный список с задачами

Плюсы *Google* задач:

- интеграция с другими сервисами *Google*, такими как Календарь, Документы и Gmail;

- быстрый и удобный интерфейс для создания, редактирования и удаления задач;

- возможность создания подзадач и отмечания их выполнения;

- доступность на всех устройствах, на которых есть доступ к интернету;

- возможность присваивания задачам даты выполнения и напоминаний;

- возможность добавления комментариев к задачам и общения с другими участниками.

Минусы *Google* задач:

- относительно простой функционал, который может не подойти для более сложных задач;

- не всегда удобно использовать на мобильных устройствах из-за необходимости доступа в интернет.

«*Todolist*: планы и задачи»

Графический интерфейс представлен на рисунках 5 - 8.

Ссылка на скачивание:

[*https://play.google.com/store/apps/details?id=com.todoist&hl=ru&gl=US*](https://play.google.com/store/apps/details?id=com.todoist&hl=ru&gl=US)



Рисунок 5 – Логотип приложения в *Google Play*

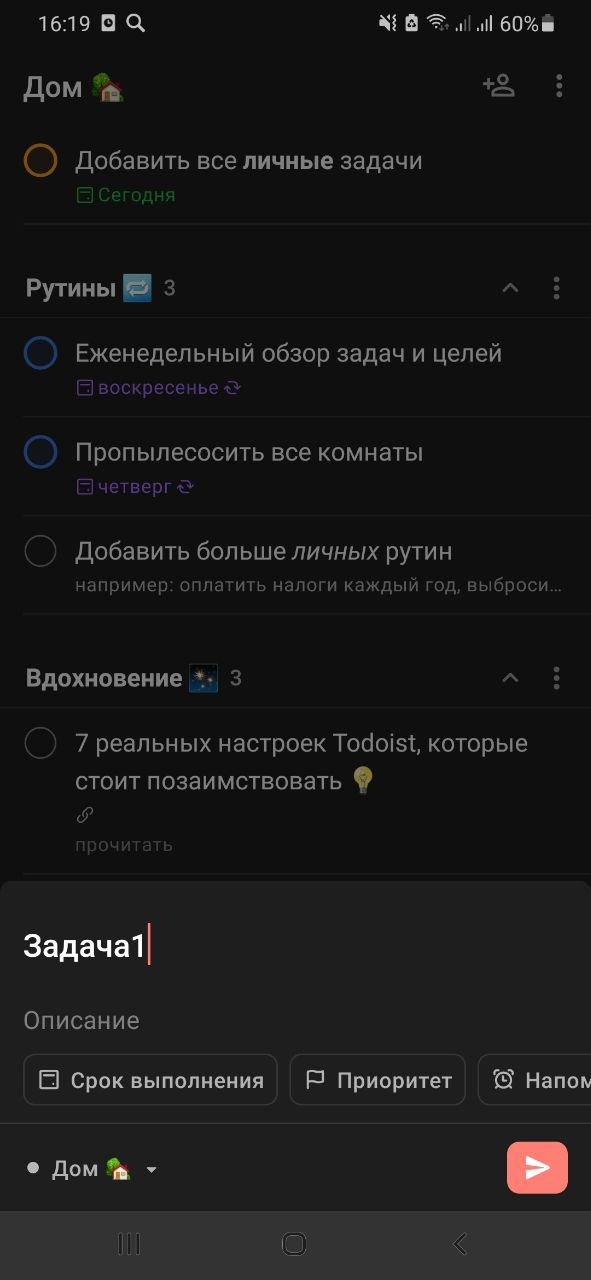


Рисунок 6 – Создание задачи

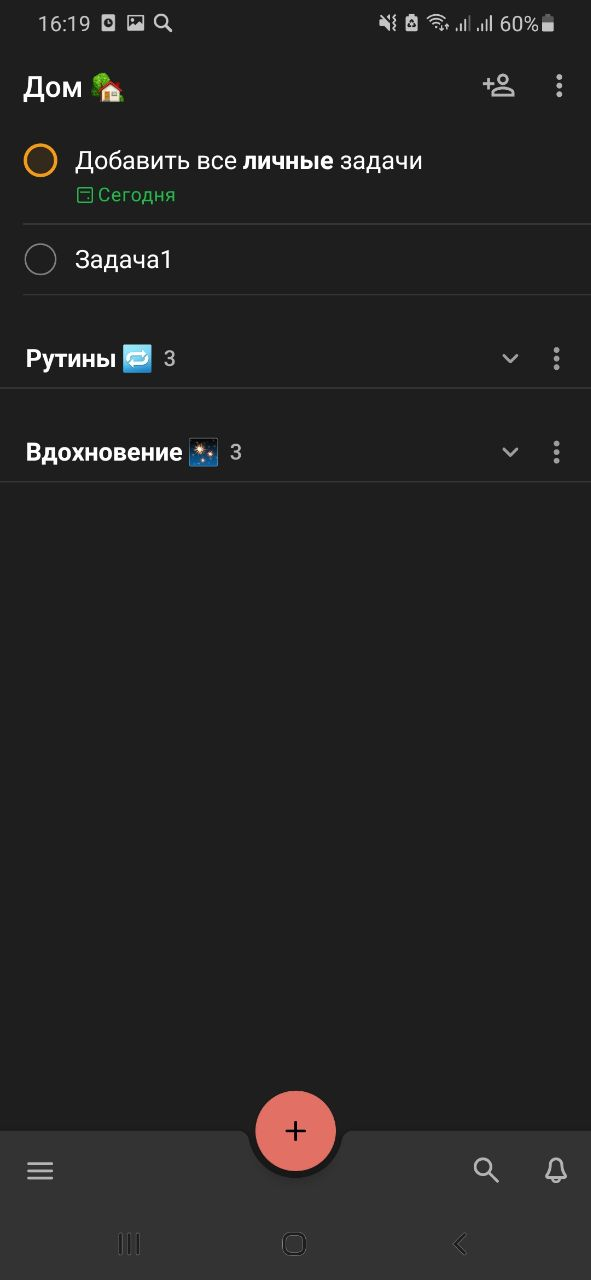


Рисунок 7 – Список с задачами



Рисунок 8 – Экран добавления других пользователей

Плюсы *Todolist*: планы и задачи:

- удобство использования и простота интерфейса;

- возможность создавать список задач и планов на каждый день, неделю или месяц.

- возможность установки приоритетов для каждой задачи;

- различные опции напоминания о задачах на выбранные даты и время;

- возможность делиться задачами с другими пользователями.

Минусы:

- необходимость подключения к Интернету для синхронизации данных, что может быть неудобно в случае отсутствия доступа к беспроводной сети;

- отсутствие возможности работы без регистрации, что может отпугнуть пользователей, которые не хотят создавать аккаунт;

- наличие платной подписки, без которой ограничивается функционал приложения.

«*TickTick*: список дел и задачи»

Графический интерфейс представлен на рисунках 9 - 13.

Ссылка на скачивание:

[*https://play.google.com/store/apps/details?id=com.ticktick.task&hl=ru&gl=US*](https://play.google.com/store/apps/details?id=com.ticktick.task&hl=ru&gl=US)



Рисунок 9 – логотип приложения в *Google Play*

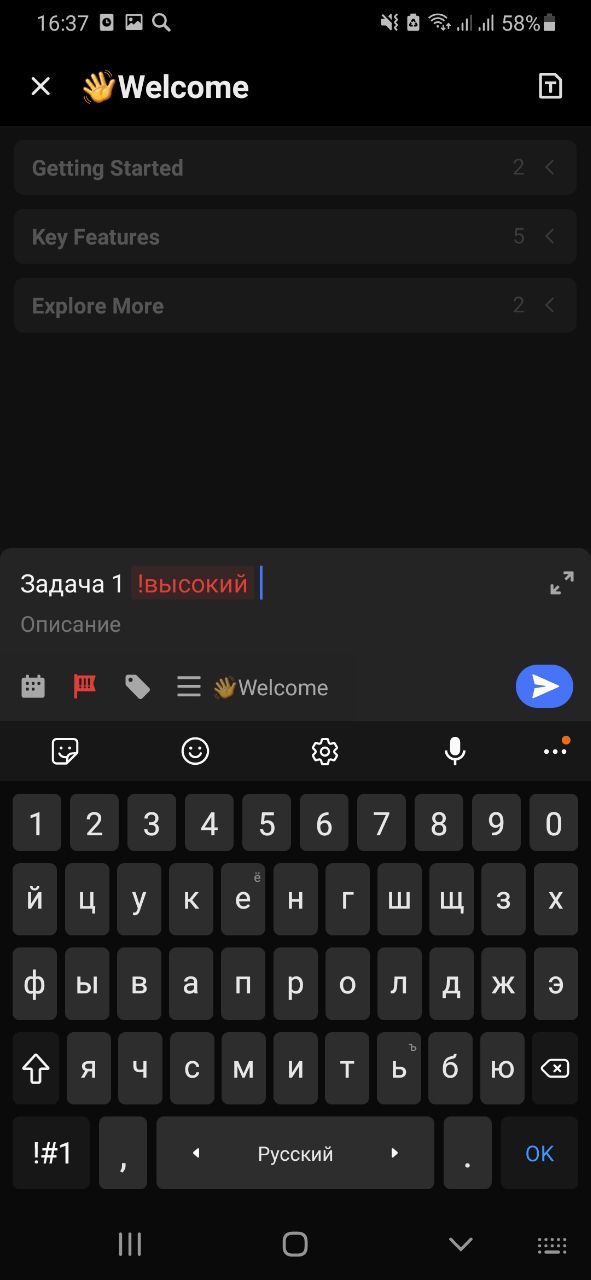


Рисунок 10 – Добавление задачи

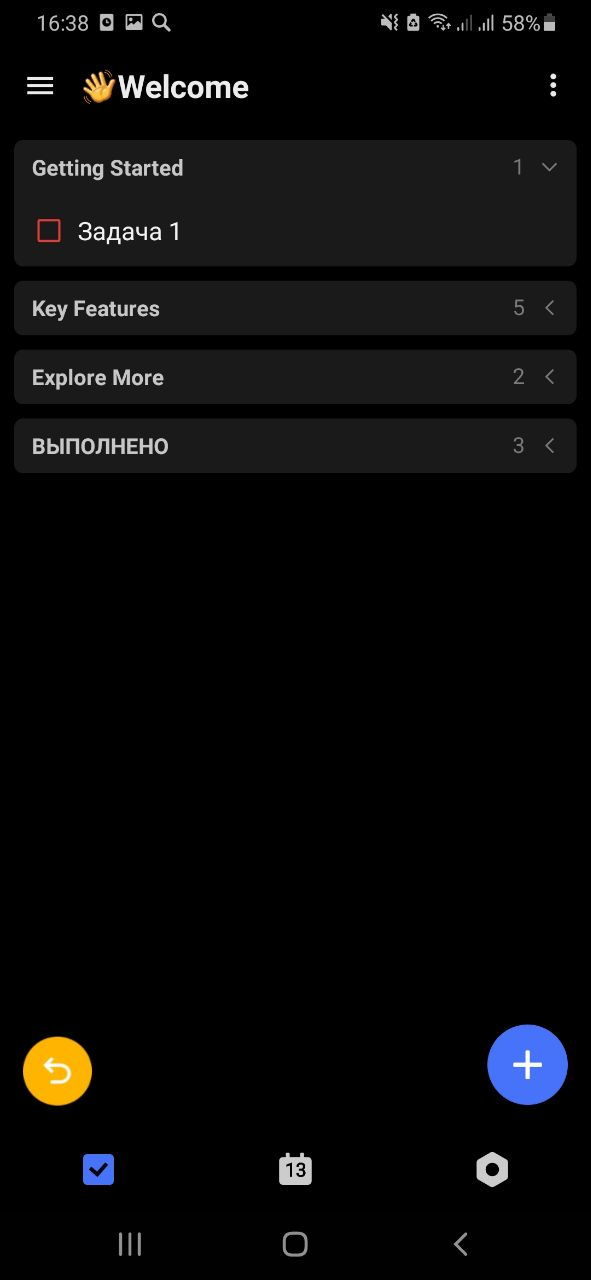


Рисунок 11 – Список задач

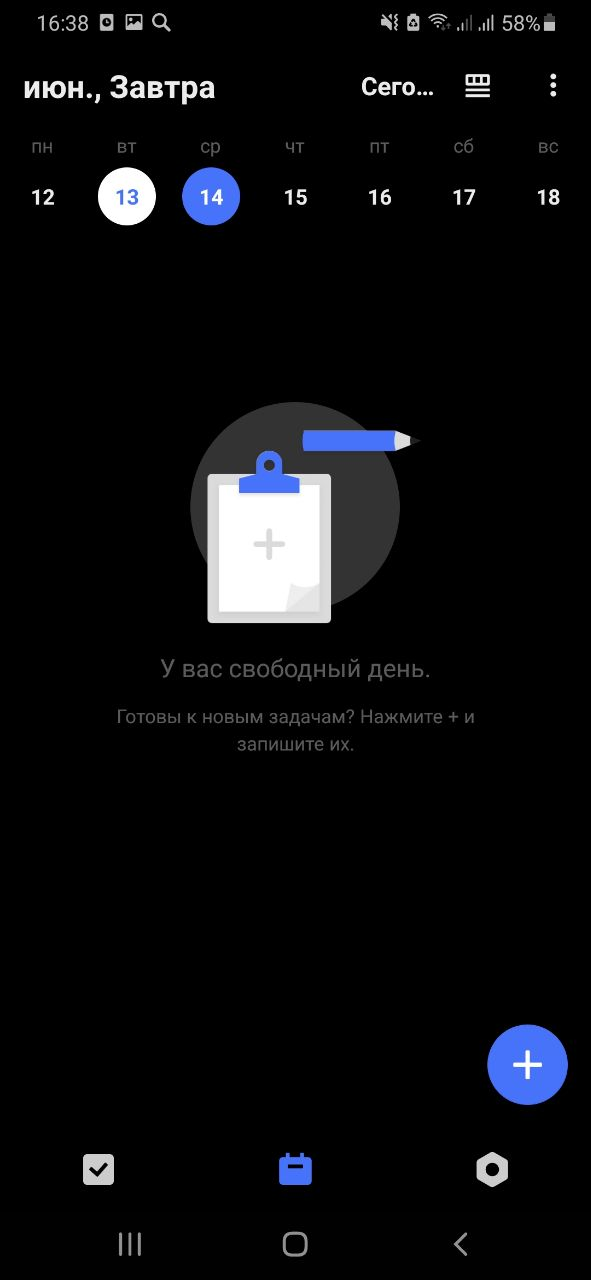


Рисунок 12 – Экран добавления задачи с помощью встроенного календаря



Рисунок 13 – Меню редактирования списка задач

Плюсы:

- имеет простой и интуитивно понятный интерфейс, который позволяет быстро создавать и управлять задачами;

- предлагает множество функций, таких как возможность установки приоритетов, напоминаний, повторения задач и т.д.;

- автоматически синхронизируется между устройствами, что позволяет использовать его на разных устройствах;

- доступно на разных платформах, включая *iOS*, *Android*, *Windows*, *Mac* и веб-браузеры.

Минусы:

- орграниченная бесплатная версия. Некоторые функции доступны только в платной версии приложения;

- нет возможности добавлять комментарии к задачам;

- ограниченное количество цветовых меток для задач. Если вам нужно использовать множество меток, вам может не хватить доступных цветов.

«*Microsoft To Do: Lists & Tasks*»

Графический интерфейс представлен на рисунках 14 - 17.

Ссылка на скачивание:

[*https://play.google.com/store/apps/details?id=com.microsoft.todos&hl=ru&gl=US*](https://play.google.com/store/apps/details?id=com.microsoft.todos&hl=ru&gl=US)



Рисунок 14 – Логотип в *Google Play*

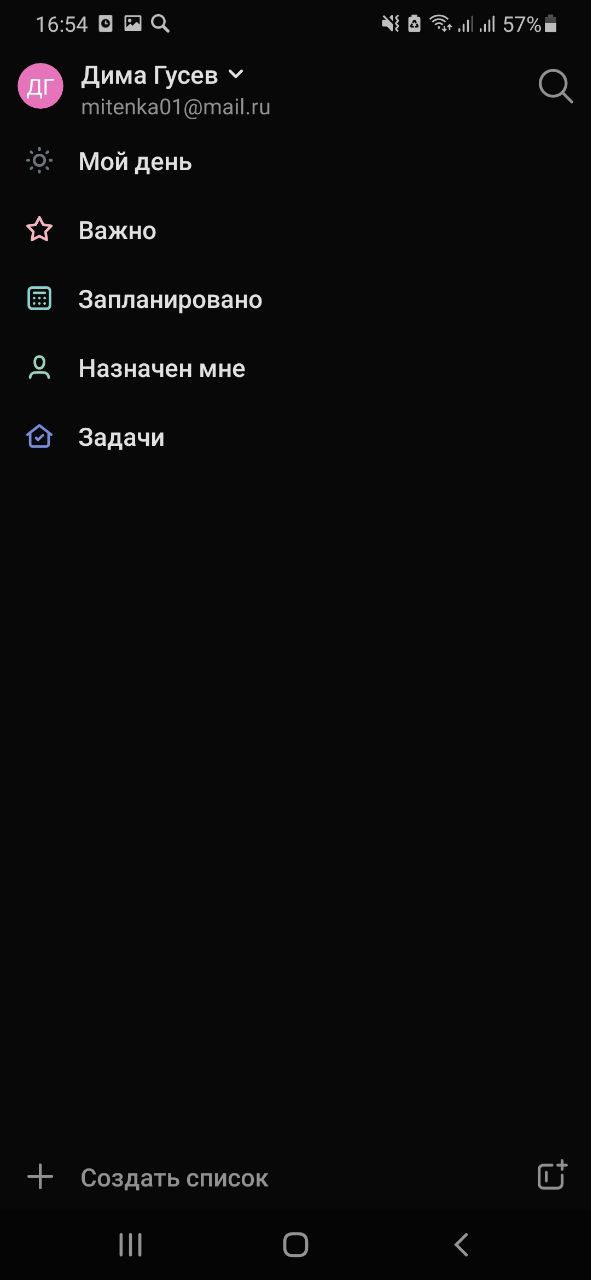


Рисунок 15 – Главное меню

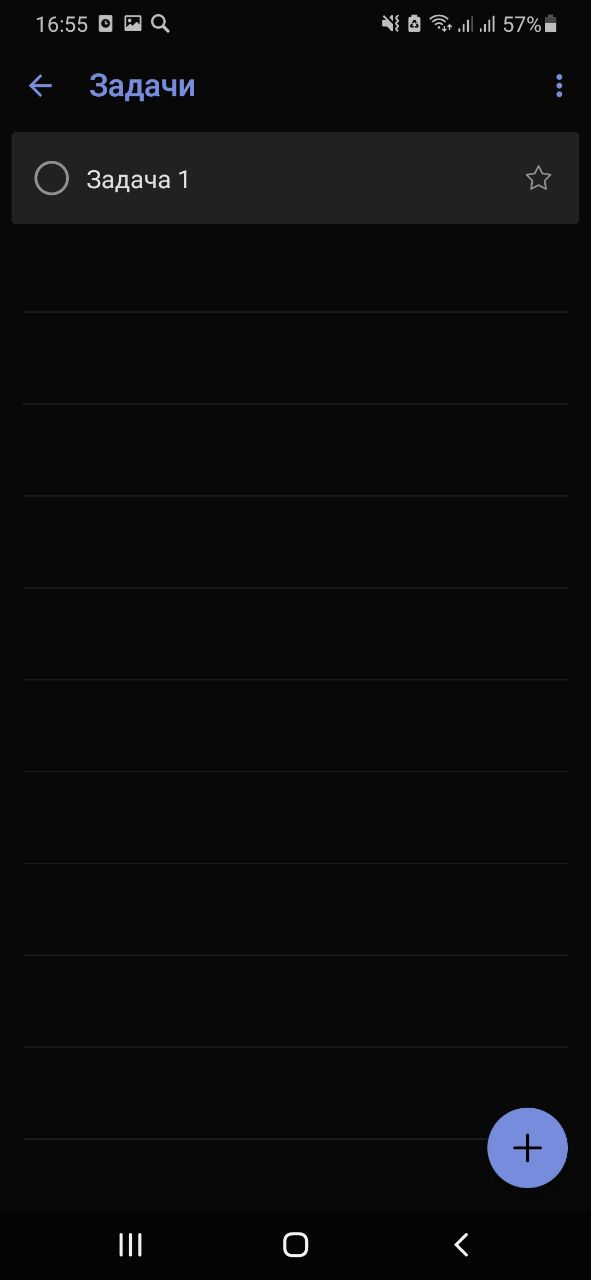


Рисунок 16 – Задача в списке



Рисунок 17 – Создание нового списка

Плюсы:

- интеграция с другими приложениями *Microsoft*, такими как *Outlook*, *OneNote*, *Teams* и другими;

- удобный и интуитивно понятный интерфейс, который позволяет легко создавать, редактировать и удалять задачи;

- возможность добавления подзадач и установки приоритетов для задач;

- автоматическая синхронизация между устройствами, что позволяет использовать приложение на нескольких устройствах;

- поддержка напоминаний и повторений задач;

- бесплатное приложение без ограничений на количество задач и списков.

Минусы:

- отсутствие некоторых функций, которые есть в других приложениях управления задачами, таких как возможность добавления комментариев к задачам или использование цветовых меток для отметки задач;

- отсутствие функции группового назначения задач, что может быть неудобно для командной работы.

## **1.1.2 Постановка задачи**

Требуется разработать приложение, которое будет соответствовать следующим целям:

- изучить основы разработки приложений под *Android*;

- разработать прототип приложения в котором может разобраться абсолютно любой пользователь;

- приложение должно быть использовано для более бытовых целей;

- разработать функцию добавления задачи;

- разработать функцию выполнения задачи;

- разработать функция напоминания задачи.

## **1.1.3 Выбор инструментов разработки**

Существует множество интегрированных сред разработки (*IDE*) для программирования для Android. Некоторые из наиболее популярных *IDE* это *Microsoft* *Visual Studio*, *Intellij IDEA*, *Android Studio*.

*Visual Studio* - это интегрированная среда разработки (IDE) от Microsoft, которая позволяет разрабатывать программное обеспечение для различных платформ, включая Android. Она обладает различными функциями и инструментами, которые позволяют упростить процесс разработки и повысить эффективность.

Достоинства *Visual Studio* для разработки на *Android*:

- интеграция с *Xamarin* - это платформа, которая позволяет разрабатывать кросс-платформенные приложения на *C#*, в том числе для *Android*. *Visual Studio* обладает нативной интеграцией с этой платформой, что позволяет разработчикам создавать приложения быстро и эффективно;

- наличие мощного отладчика. *Visual Studio* обладает мощным и удобным отладчиком, который позволяет быстро находить и исправлять ошибки, а также отслеживать производительность приложения;

- интеграция с *Git* - *Visual Studio* поддерживает интеграцию с системой контроля версий *Git*, что позволяет разработчикам упростить процесс работы с исходным кодом приложения.

Недостатки *Visual Studio* для разработки на *Android*:

- высокая стоимость. *Visual Studio* является платным продуктом, что может быть недоступно для многих начинающих разработчиков или команды, которые не могут позволить себе высокие затраты на разработку;

- не подходит для разработки на *Java*. *Visual Studio* поддерживает разработку приложений на *Android* на языке программирования *C#*, что не является самым популярным языком программирования для разработки приложений на *Android*;

- высокие требования к аппаратному обеспечению. *Visual Studio* может потребовать высокопроизводительное аппаратное обеспечение, особенно при работе с крупными проектами. Это также может потребовать дополнительных затрат.

В целом, для разработки кросс-платформенных приложений на *C#* для *Android*, *Visual Studio* - отличный выбор. Но если необходимо разработать программу на Java или не хватает средств для покупки *IDE*, то необходимо рассмотреть другие варианты.

*IntelliJ IDEA* - это интегрированная среда разработки (*IDE*), выпущенная компанией *JetBrains*, которая предназначена для разработки программного обеспечения на различных языках программирования, включая *Java*, *Kotlin*, *C++,* *JavaScript* и многие другие. Она также поддерживает разработку приложений на *Android* и обладает множеством функций и инструментов, которые позволяют упростить процесс разработки и улучшить качество и производительность приложений.

Достоинства *IntelliJ* *IDEA* для разработки на *Android*:

- поддержка языка *Kotlin*. *Kotlin* является официальным языком для разработки приложений на *Android*. *IntelliJ* *IDEA* обладает интеграцией с *Kotlin* и позволяет легко и быстро создавать приложения на этом языке;

- интеграция с системой сборки *Gradle*. *IntelliJ* *IDEA* имеет нативную интеграцию с системой сборки *Gradle*, которая используется для сборки и компиляции приложения. Это упрощает процесс сборки приложения и облегчает автоматизацию;

- мощный отладчик. *IntelliJ* *IDEA* имеет мощный и удобный отладчик, который позволяет быстро находить и исправлять ошибки в приложении.

Недостатки *IntelliJ* *IDEA* для разработки на *Android*:

- стоимость. *IntelliJ* *IDEA* является платной *IDE*, что может быть недоступно для многих начинающих разработчиков или малых команд, которые не могут позволить себе высокие затраты на разработку;

- высокие требования к аппаратному обеспечению. *IntelliJ* *IDEA* может потребовать высокопроизводительное аппаратное обеспечение для работы, особенно при работе с крупными проектами;

- отсутствие поддержки для некоторых библиотек. *IntelliJ* *IDEA* не обладает поддержкой для некоторых библиотек, которые могут быть полезны для разработки на *Android*, и требуют дополнительной работы для их интеграции.

В целом, для разработки приложений на *Java* или *Kotlin* для *Android* *IntelliJ* *IDEA* - отличный выбор. Интеграция с системой сборки *Gradle*, мощный отладчик, большой выбор инструментов и интуитивно понятный интерфейс делают эту *IDE* одной из наиболее популярных на рынке разработки на *Android*. Но если необходимо разработать программу на другом языке или не хватает средств для покупки *IDE*, то необходимо рассмотреть другие варианты.

*Android Studio* - это официальная интегрированная среда разработки для Android, созданная компанией *Google*. Инструмент написан на базе среды разработки *IntelliJ* *IDEA* и предназначен для разработки мобильных приложений на платформе *Android*. Он имеет множество функций и инструментов, которые делают процесс разработки более эффективным и удобным.

Достоинства *Android* *Studio* для разработки на *Android*:

- набор инструментов. *Android* *Studio* обладает большим количеством инструментов, которые облегчают и ускоряют процесс разработки, включая макет-редактор, богатый набор компонентов, инструменты анализа и тестирования [1];

- интеграция с *Android* *SDK*. *Android* *Studio* имеет нативную интеграцию с *Android* *SDK*, что позволяет легко и быстро создавать приложения для устройств под управлением *Android*;

- наличие эмулятора *Android*. *Android* *Studio* включает в себя эмулятор *Android*, который позволяет разработчикам проверять приложения на различных устройствах, включая самые новые и свежие [2];

- интеграция с системой версионного контроля. *Android* *Studio* интегрируется с системой контроля версий *Git*, что позволяет разработчикам легко управлять и обновлять исходный код в различных ветках;

- бесплатность - *Android* *Studio* является бесплатным продуктом, что делает его доступным для большинства пользователей.

Недостатки *Android* *Studio* для разработки на *Android*:

- высокие требования к аппаратному обеспечению - *Android* *Studio* требует высокопроизводительное аппаратное обеспечение для работы, особенно при работе с крупными проектами;

- сложность в использовании. Как и любая другая *IDE*, *Android* *Studio* может быть сложна в использовании для новичков в *Android*-разработке, требуя некоторого опыта и знаний программирования;

- ошибки эмулятора. Хотя эмулятор *Android* *Studio* предлагает отличный способ проверить приложение на различных устройствах, он иногда может иметь ошибки или быть медленным [3].

*Android* *Studio* является лучшим выбором для разработки приложений на платформе *Android*, благодаря интуитивно понятному интерфейсу, большому количеству инструментов и поддержке *Android* *SDK*. Несмотря на высокие требования к аппаратному обеспечению и сложность использования, *Android* *Studio* является стандартной *IDE* для *Android*-разработки.

## **1.1.4 Выбор языка программирования**

Для разработки приложения для *Android* можно писать на нескольких языках программирования, таких как Java, *Kotlint*.

*Kotlin* - это статически типизированный язык программирования, который совместим с технологиями *Java*. Он был создан компанией *JetBrains* и быстро завоевал популярность в мире *Android*-разработки благодаря своим множеством особенностей и преимуществам по сравнению с *Java*.

Основные особенности *Kotlin* в разработке под *Android*:

- понятный синтаксис. *Kotlin* обладает более кратким и легким для понимания синтаксисом, который делает язык простым в изучении и использовании;

- наличие *Null* *Safety*. *Kotlin* имеет функциональность *Null* *Safety*, которая позволяет избежать ошибок, связанных с *null*-объектами. Это делает код более читабельным и уменьшает риск ошибок по мере развития приложения;

- высокая производительность. *Kotlin* обладает многими функциями, которые позволяют достичь высокой производительности приложений, включая поддержку многопоточности, лямбда-функций и инлайновых функций;

- активное развитие. *Kotlin* продолжает развиваться и улучшаться с каждым выпуском новой версии компанией *JetBrains*, что означает, что в будущем для *Android*-разработки ожидается большое количество новых функций и возможностей.

*Java* является одним из основных языков программирования, используемых для разработки приложений под *Android*. Основные особенности *Java* в разработке под *Android* включают:

- объектно-ориентированный подход. *Java* поддерживает программирование в стиле ООП, что позволяет разрабатывать сложные приложения с использованием многих классов и объектов [4];

- богатая библиотека. *Java* имеет множество библиотек, которые упрощают разработку приложений, таких как библиотека для работы с графикой, базами данных и т.д.;

- многопоточность. *Java* имеет встроенную поддержку многопоточности, что позволяет приложениям выполнять множество задач одновременно, повышая производительность и отзывчивость;

- простота. *Java* имеет простой синтаксис и легко читаемый код, что упрощает разработку и поддержку приложений [5].

## **1.1.5 Выбор решения для хранения данных**

Для хранения данных можно использовать несколько решений, таких как БД *SQLite*, библиотека *Room*.

*SQLite* - это компактная реляционная БД, которая предлагает полный набор функций для работы с *SQL*, таких как создание таблиц, выполнение запросов, индексирование и транзакции. *SQLite* интегрирована в платформу *Android* и может быть использована для хранения и управления данными внутри *Android*-приложений.

Основные особенности *SQLite* включают:

- надежность. *SQLite* позволяет сохранять данные в очень надежном формате, что позволяет убедиться, что данные не будут потеряны или повреждены;

- гибкость. *SQLite* поддерживает широкий спектр типов данных и индексирование, что позволяет создавать сложные таблицы;

- безопасность. *SQLite* поддерживает многопоточность и транзакции, что позволяет избежать конфликтов при одновременном доступе к данным;

- поддержка. *SQLite* является широко используемой и хорошо поддерживаемой базой данных, что обеспечивает достаточный уровень поддержки и ресурсов для работы с этой базой данных.

*Room* - это *persistence*-библиотека для *Android*, которая создана на основе *SQLite*. Он позволяет легко и эффективно работать с базой данных в *Android*-приложениях, предоставляя удобный *API* для запросов к базе данных и обработки результатов.

Основные особенности *Room* включают:

- аннотации. *Room* использует аннотации для определения сущностей, которые будут сохраняться в базу данных, а также аннотации для определения запросов к базе данных. Это позволяет использовать *API* для запросов к базе данных без необходимости написания *SQL*-запросов вручную;

- поддержка *LiveData*. *Room* предоставляет поддержку живых данных, которые автоматически обновляются при любых изменениях в базе данных. Это значит, что *LiveData* может использоваться для автоматической перезагрузки данных из базы данных при изменении данных;

- поддержка *RxJava* - *Room* поддерживает *RxJava*, что позволяет работать с базой данных на функциональном уровне. Это упрощает написание чистого и реактивного кода;

- асинхронное выполнение запросов. *Room* позволяет выполнять запросы к базе данных асинхронно, что позволяет избежать блокировки потока *UI*;

- настроен для использования с архитектурой компонентов. *Room* разработан, чтобы легко интегрироваться с архитектурой компонентов *Android*, такой как *ViewModel* и *LiveData*.

## **1.1.6 Выводы**

По итогу проведенных анализов для разработки Андроид-приложения «Планировщик задач» были выбраны среда разработки *Android* *Studio*, язык программирования *Java* и библиотека для работы с БД *Room*.

## **1.2 Реализация приложения**

## **1.2.1 layouts**

Графический интерфейс приложения реализован с помощью 3-х файлов: *activity\_main.xml, add\_activity.xml, list\_row\_main.xml.*

*list\_row\_main.xml* – отдельная настройка внешнего вида задачи на главном экране. Содержит поле для отображения названия задачи и кнопу для её выполнения.

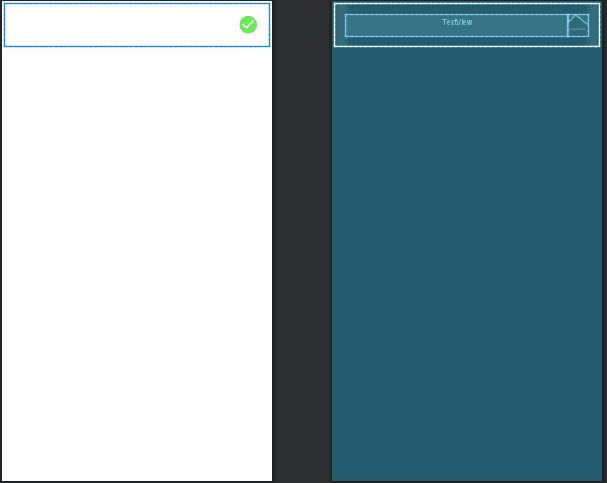


Рисунок 18 – *list\_row\_main.xml*

*activity\_main.xml* – главная активити, с которой начинается запуск приложения. На ней находятся кнопка добавления новой задачи и список из существующих задач

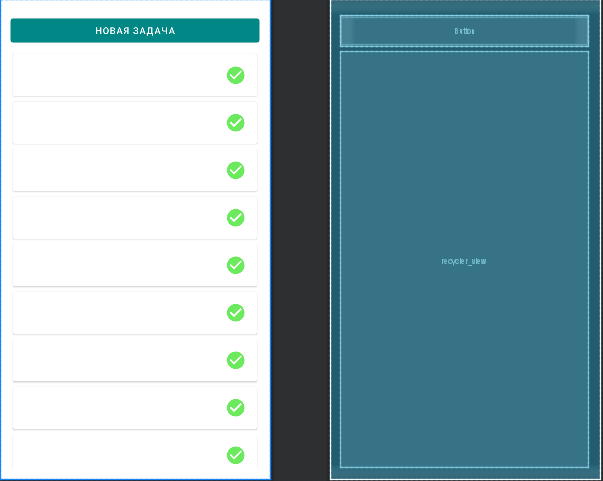


Рисунок 19 – *activity\_main.xml*

*add\_activity.xml* – содержит внешний вид экрана для добавления новой задачи. Содержит текстовые поля для ввода названия, даты и времени задачи, кнопку их добавления и чекбокс для настройки о необходимости напоминания о задаче.

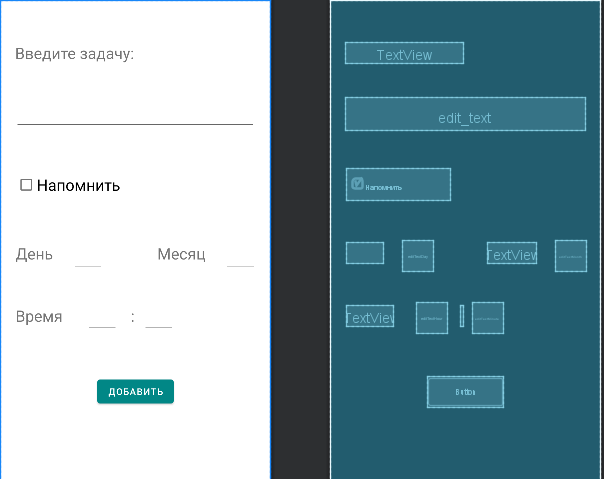


Рисунок 20 *– add\_activity.xml*

## **1.2.2 java-классы для работы с БД**

*MainDao.java* – содержит основные запросы:

- *insert* – добавление нового поля;

- *delete* – удаление поля;

- *getAll* – получение всех полей БД.

*MainData.java* – инициализирует поля БД и содержит их свойства *get*() и *set*().

*RoomDB –* инициализирует БД.

Таким образом мы имеем БД с названием «*database*» с полями:

- *ID* типа *integer*;

- *text* типа *string*.

## **1.2.3 java-классы для работы с активити**

*MainActivity.java* – главная активити, с которой начинается запуск приложения. Весь отображаемый контент получает их файла activity\_main.xml. Из основных функций содержит:

- *onCreate*() – один из жизненных циклов активити. Загружает весь контент из *xml*-файла. При необходимости принимает аргументы из другой активити для добавления задачи;

- *StartNewActivity*() – запускает новою активити *AddTask*;

- *addTask*() – добавляет новую задачу.

*MainAdapter.java* – адаптер для оформления списка задач. Также отвечает за логику кнопки завершения задачи. Также получает контент с *list\_row\_main.xml*.

*AddTask*.*java* – Активити для добавления новой задачи. Содержит функции:

-*OnCreate*() – один из жизненных циклов активити. Загружает весь контент из xml-файла. Содержит основную логику чекбокса, который делает активными или неактивными поля для ввода даты и времени;

-*StartNewActivity*() - запускает *MainActivity*, передаёт в неё название задачи, при активном чекбоксе также передаёт дату и время.

*AlarmReceiver.java* – описывает уведомление, всплывающее когда необходимо напомнить о задаче.

## **1.2.4 Выводы**

Была выбрана среда *Android* *Studio* для написания приложения. Написан код на языке *Java*. Реализованы все требования к поставленной задаче.

## **1.3 Тестирование приложения**

## **1.3.1 Тестирование**

Для тестирования использовалось устройство *Samsung Galaxy A50*. Оно было подключено через *USB* к ПК и среда распознала его (рисунок 21).



Рисунок 21 – Среда распознала устройсвто

Был осуществлён запуск приложения. На устройстве появился соответствующий значок приложения (рисунок 22).

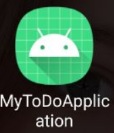


Рисунок 22 – Значок приложения

При запуске приложения появляется главный экран (рисунок 23).



Рисунок 23 – Главное меню

При нажатии на кнопку новая задача появляется меню для добавления задачи (рисунок 24).

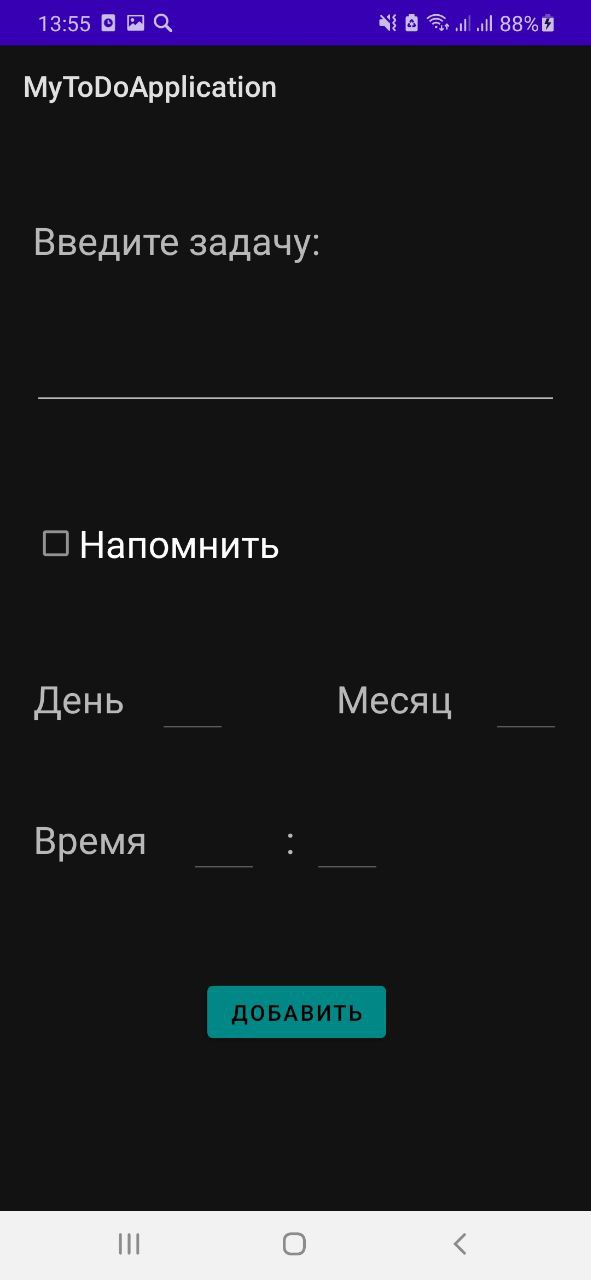


Рисунок 24 – Меню добавления задачи

Вводим название задачи. При неактивном чекбоксе нельзя ввести дату и время. Нажимаем кнопку добавить (рисунок 25).

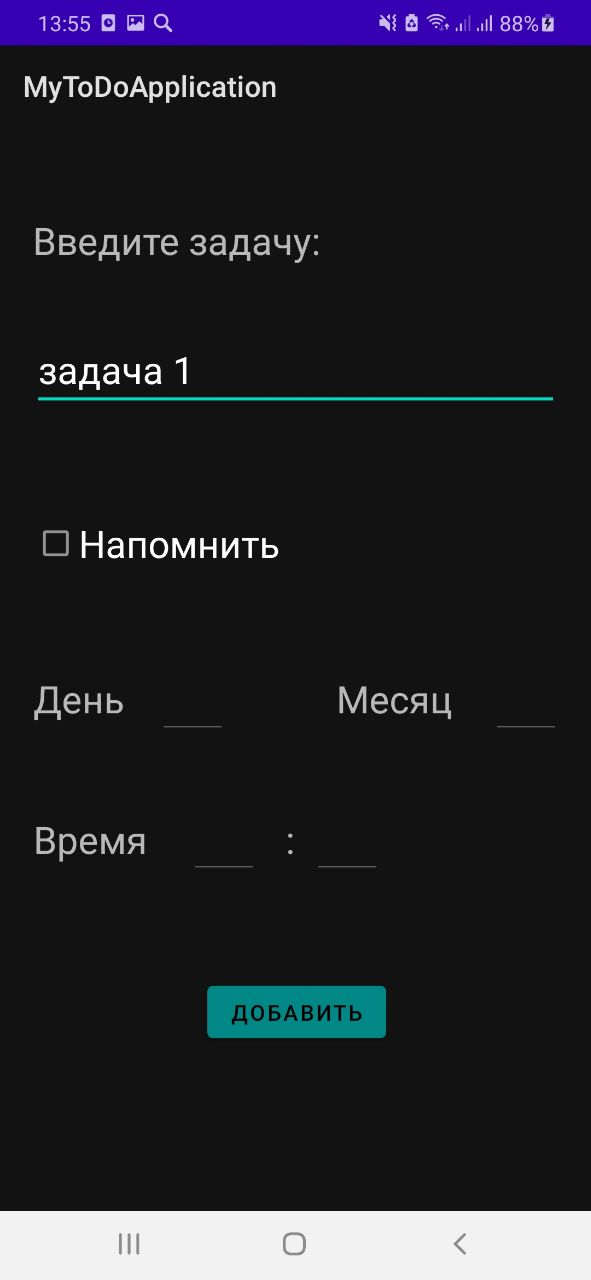


Рисунок 25 – Заполненное поле

После нажатия на кнопку в главном меню видим, что задача успешно добавилась (рисунок 26).



Рисунок 26 – Задача добавилась

Теперь введем дату и время (рисунок 27).

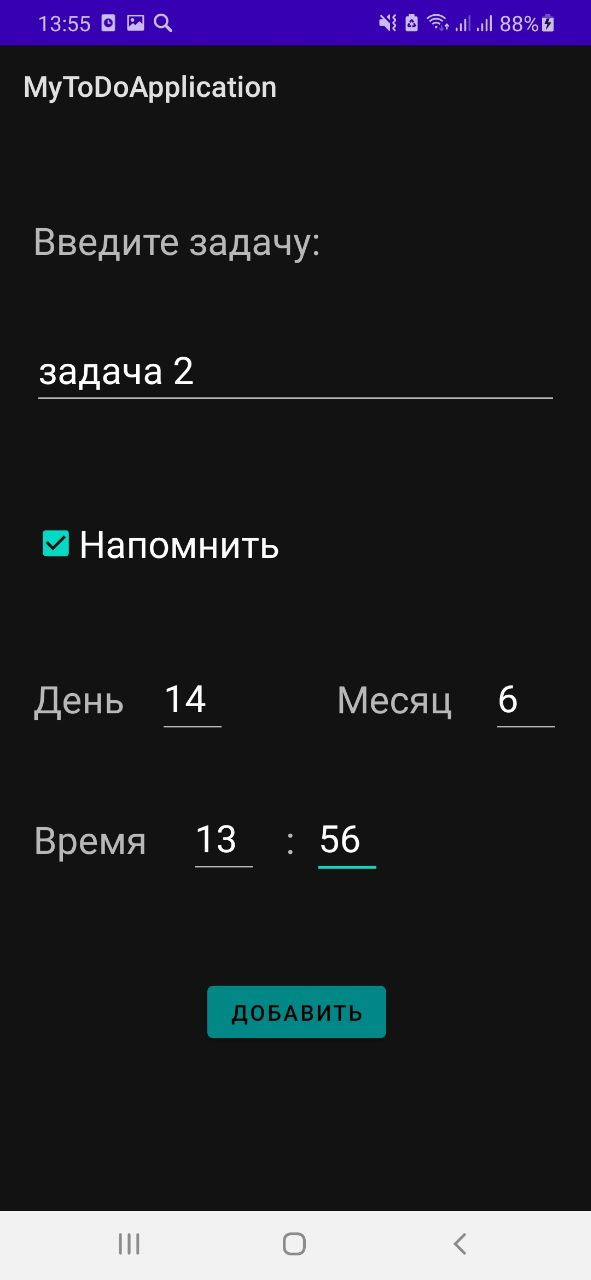


Рисунок 27 – Заполненные поля

После нажатия на кнопку в главном меню видим, что задача также успешно добавилась (рисунок 28).

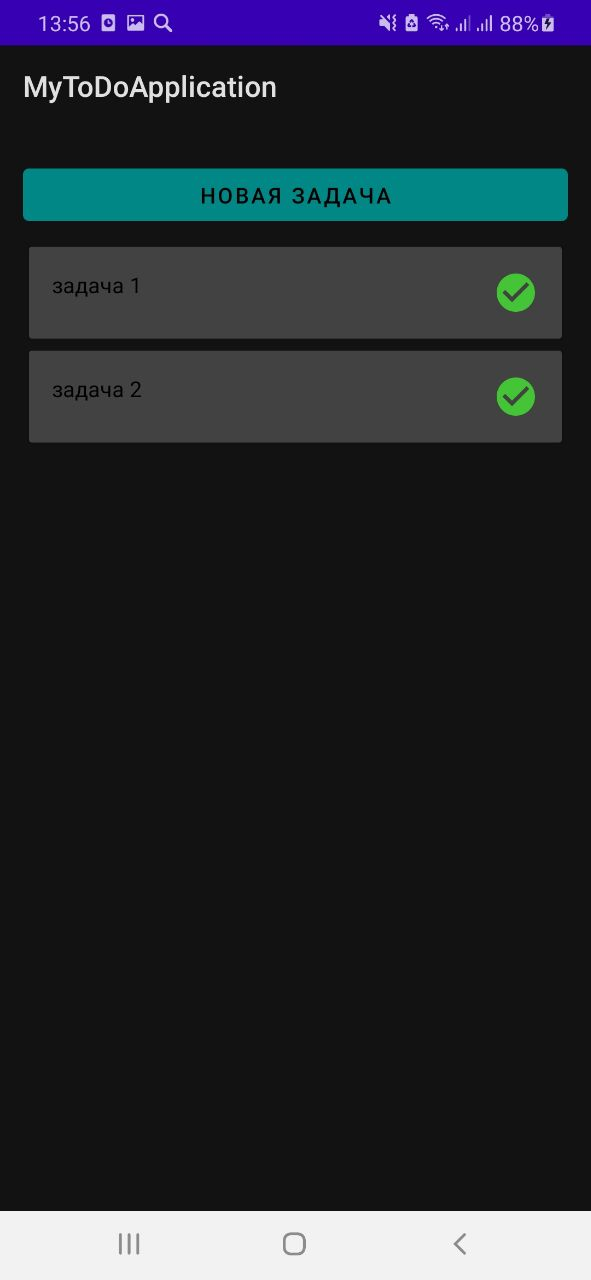


Рисунок 28 – Добавленные задачи

Когда наступает введенное время, то на устройство приходит уведомление (рисунок 29).

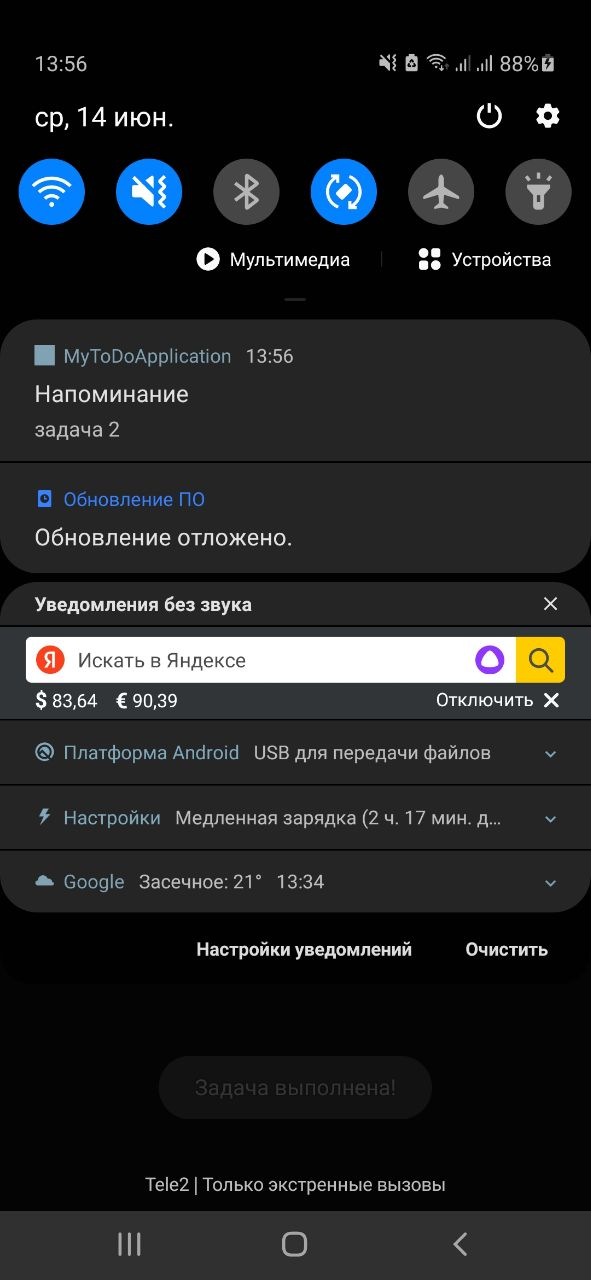


Рисунок 29 – Уведомление

Теперь добавим ещё одну задачу (рисунок 30).

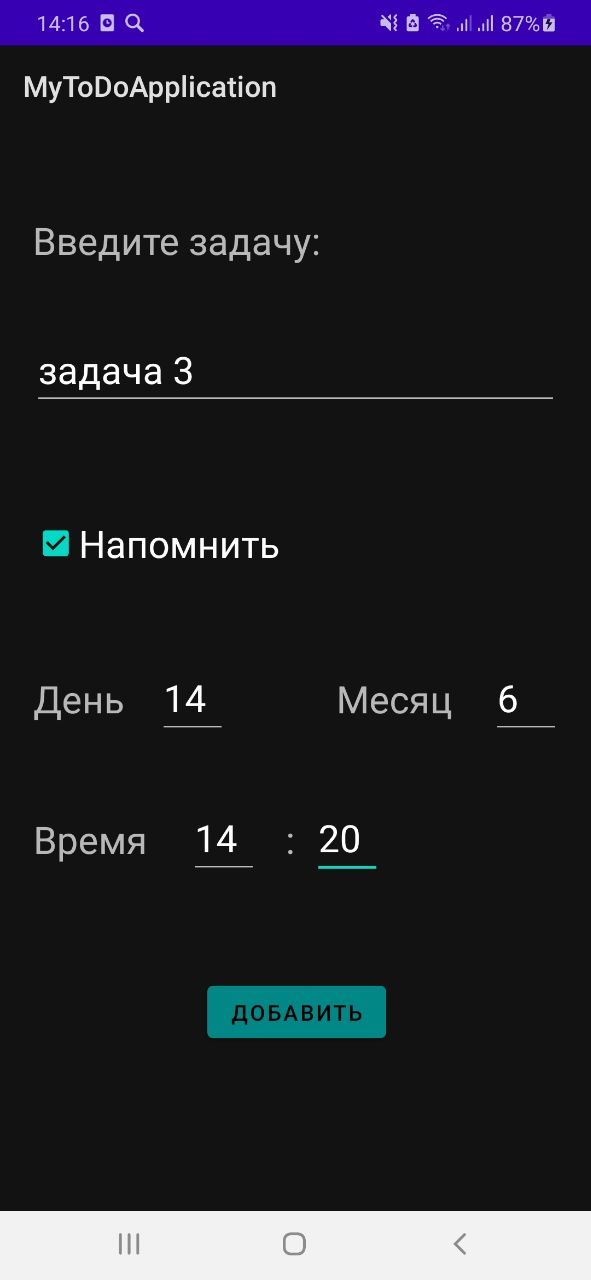


Рисунок 30 – Заполненные поля

Попробуем ее выполнить. При нажатии на соответствующую икнонку всплывает окно с подтверждением (рисунок 31).

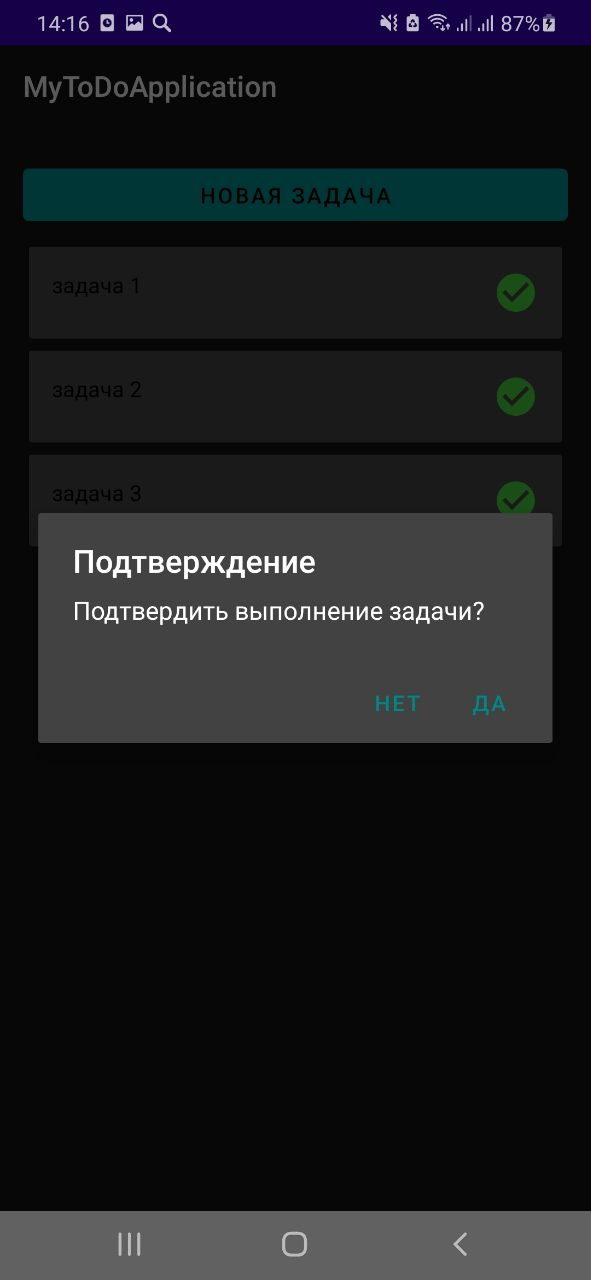


Рисунок 31 – Окно подтверждения

Если нажать «нет», то ничего не произойдёт, если ответить «да», то задача удалиться из списка и появится уведомление о её выполнении. При этом уведомление о напоминании задачи не появится (рисунок 32).

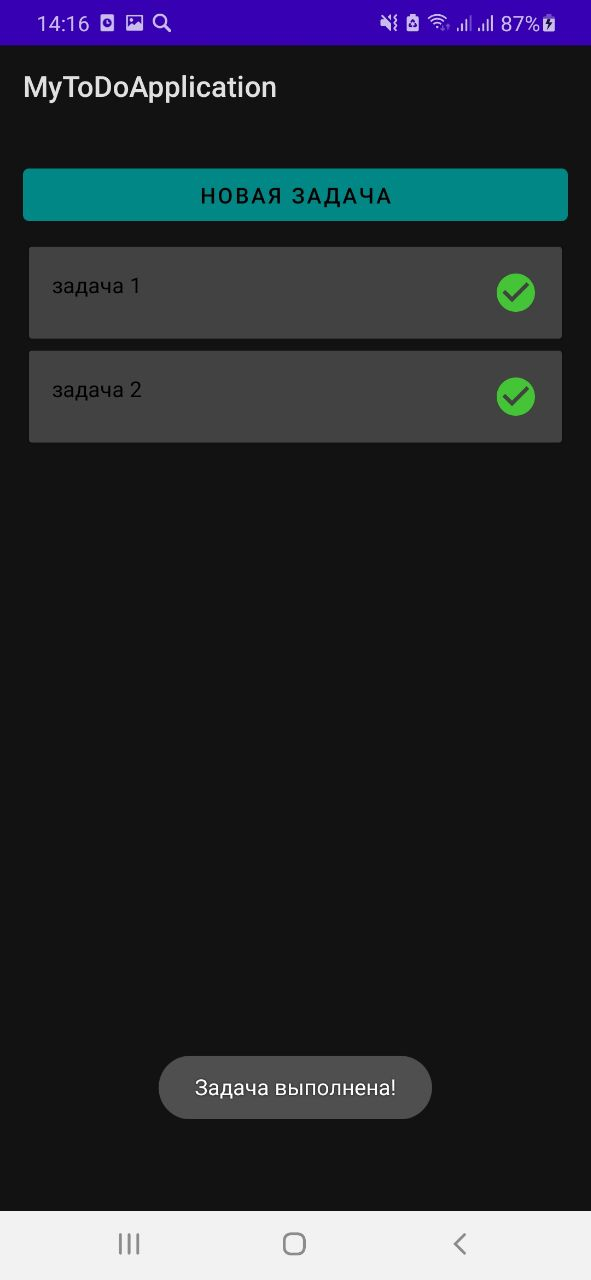


Рисунок 32 – Задача выполнена

## **1.3.2 Выводы**

Использование приложения «Планировщик задач» на физических мобильных устройствах не выявило ошибок и проблем во время работы. Тесты показали, что приложение работает корректно и может использоваться пользователем.

## **2 Безопасность жизнедеятельности**

## **2.1 Анализ негативных факторов**

При работе на компьютере человек сталкивается с несколькими негативными факторами: длительное сидение в одной позе, ограниченная подвижность, перенапряжение глаз. Эти факторы могут привести к утомлению, снижению производительности и общей усталости. Чтобы избежать негативного влияния компьютера на здоровье, необходимо соблюдать ряд правил, включая организацию рабочего места, обеспечение достаточной освещенности и соблюдение правил безопасности. Отдых и работа должны быть регулированы.

## **2.1.1 Оборудование рабочего места**

Для оптимального рабочего места важно соблюдать несколько основных требований. Расстояние между глазами и экраном должно быть в пределах 600-700 мм. Экран должен быть расположен ниже уровня глаз и наклонен на 10-15 градусов. Рабочий стол должен регулироваться в высоте в пределах 680-800 мм, а в случае отсутствия такой опции, быть на уровне 725 мм. Клавиатура должна быть расположена на расстоянии 100-300 мм от пользователя или на специальной регулируемой по высоте поверхности. Рабочий стол должен предоставлять достаточное пространство для ног на разных уровнях, включая высоту, глубину и ширину. На рисунке 33 изображена организация рабочего места, которая отвечает указанным требованиям.

Для рабочего места с дисплеем важно обеспечить не только удобство оператора в положении сидя, но и избежать перегрузки костно-мышечной системы.

Организовав правильно рабочее место, также необходимо обеспечить правильную посадка пользователя ПК, правильная посадка изображена на рисунке 34.

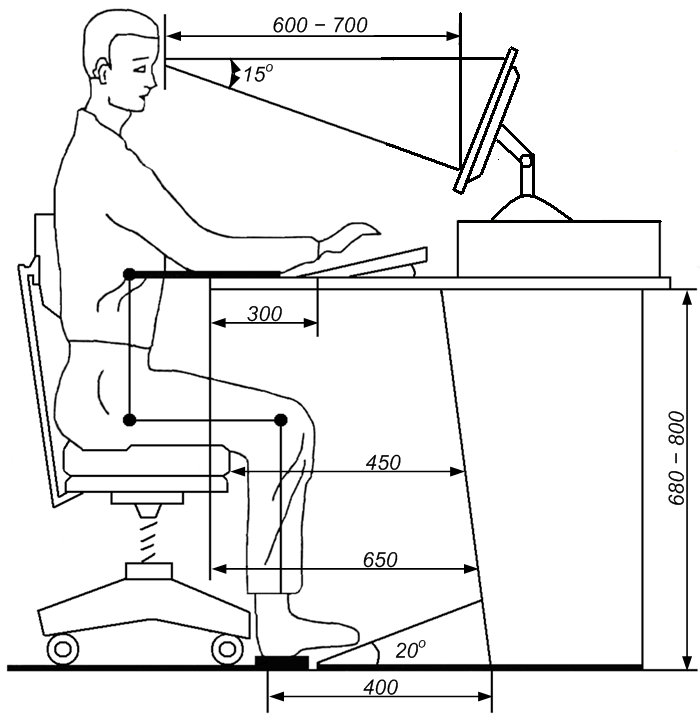


Рисунок 33 – Организация рабочего места

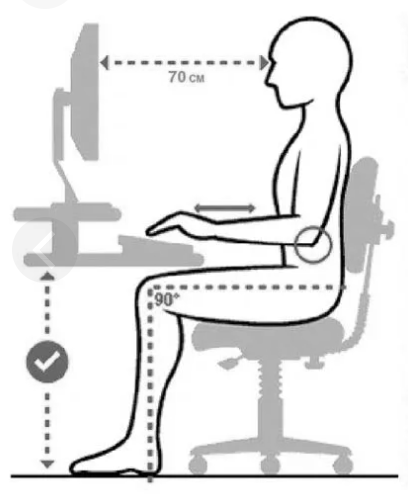


Рисунок 34 – Правильная посадка за ПК

Чтобы обеспечить правильную посадку пользователя ПК, необходимо следить за несколькими условиями. Запястья следует держать в прямом положении, расслабить плечи, а локти должны лежать на столе под прямым углом (90-100 градусов).

## **2.1.2 Освещенность рабочего места**

Стандарты освещенности учитывают характер работы, место и поверхность. Для пультов ЭВМ и дисплеев рекомендуется использовать комбинированное освещение не менее 400 люкс и общее освещение не менее 200 люкс. Важно обеспечить равномерную яркость на рабочем месте, избегая ярких и блестящих предметов. Рекомендуется использовать яркие поверхности для разнообразия в поле зрения. Обычно для освещения используются потолочные или встроенные светильники с люминесцентными лампами. Для избежания засветки экранов, светильники общего освещения следует установить сбоку от рабочего места, параллельно линии зрения оператора.

## **2.1.3 Электробезопасность**

Соответствующие требования к электробезопасности определены в ГОСТ 12.1.019-2017 "ЭЛЕКТРОБЕЗОПАСНОСТЬ". Работа ПК от сети переменного тока 50 Гц с напряжением 230 В представляет риск для здоровья и жизни. Чтобы гарантировать безопасность, при работе на ПК необходимо соблюдать ряд требований. Все устройства должны быть подключены к одной фазе электрической сети, а питание устройств ПК должно осуществляться через розетки типа "Европа" с заземляющими контактами. Кроме того, все розетки должны быть маркированы по напряжению, а номинальное напряжение сети должно быть нанесено на стену или щит крупными символами яркой краской (высотой не менее 50 мм) возле розетки. Важно проявить осторожность при использовании вычислительной техники и периферийного оборудования в отношении электропроводки и приборов, чтобы избежать угрозы для здоровья и жизни. Необходимо периодически проверять исправность электропроводки, выключателей и розеток,

используемых для подключения оборудования. При обнаружении неисправностей следует немедленно отключить электрооборудование до тех пор, пока все неисправности не будут устранены.

## **2.1.4 Пожарная безопасность**

Пожары, которые возникают во время использования электроустановок и ПК, чаще всего вызваны коротким замыканием проводов и устройств, возгоранием горючих материалов рядом с электроустановками и большими переходными сопротивлениями в контактных соединениях. Чтобы обеспечить безопасность людей, предотвратить возникновение и распространение пожаров и создать условия для их эффективной тушения, необходимо принимать организационные меры в соответствии с федеральным законом "О пожарной безопасности". Для улучшения пожаробезопасности помещений следует осуществлять следующие меры: не загромождать проходы и запасные выходы, обеспечить доступность средств пожаротушения, оснастить систему вентиляции устройствами, автоматически отключающими ее при пожаре и огнезадерживающими клапанами, проверять и поддерживать в исправном состоянии автоматическую пожарную сигнализацию и средства пожаротушения в соответствии с типовыми правилами технического содержания установок пожарной автоматики.

## **2.1.5 Режим труда и отдыха**

Сотрудник, которому была поручена работа с ПК, должен придерживаться правильного режима труда и отдыха, соответствующего продолжительности, типу и категории своей трудовой деятельности. Работы, связанные с использованием персонального компьютера, разделены на три категории, в зависимости от цели:

- группа А- чтение информации с экрана видеодисплейного терминала с предварительным запросом;

- группа Б - ввод информации;

- группа В- творческая деятельность в режиме диалога с ПК.

Основная деятельность, связанная с ПК, определяется как та, которая занимает не менее 50% всего рабочего времени в течение рабочего дня или смены. Для каждого типа трудовой деятельности установлены три уровня интенсивности и напряженности:

- группа А - уровень категории зависит от общего числа считываемых символов за рабочую смену (не более 60 000 знаков за смену);

- группа Б - уровень категории зависит от общего числа считанных или введенных символов за рабочую смену (не более 40 000 знаков за смену);

- группа В - уровень категории зависит от общего объема времени, потраченного на непосредственную работу с ПК за рабочую смену (не более 6 часов за смену).

Таблица 1.1 отображает общее время перерывов для каждой категории и вида работы.

Таблица 1.1 – Время перерывов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категория работы с ПЭВМ или ВДТ | Уровень нагрузки за рабочую смену при видах работы с ВДТ | | | Суммарное время регламентированных перерывов, мин | |
| группа А, количество знаков | группа Б, количество знаков | группа В час | при 8-часовой смене | при 12-часовой смене |
| 1 | До 20000 | До 15000 | До 2 | 50 | 80 |
| 2 | До 40000 | До 30000 | До 4 | 70 | 110 |
| 3 | До 60000 | До 40000 | До 6 | 90 | 140 |

## **Заключение**

В ходе выполнения выпускной квалификационной работы было разработано Андроид-приложение «Планировщик задач» в *Android Studio*, с использованием языка *Java*, как основного языка программирования.

Разработанное приложение позволяет записывать задачи и уведомлять о их выполнении.

Одним из достоинств разработанного приложения является максимальная простота интерфейса, в котором разберётся любой пользователь; легковесность приложения, что позволяет запускать его на любых устройствах.

В дальнейшем можно доработать разработанное приложение, добавив маркер для важных задач для вывода их в более высокое положения, списки задач и возможность их редактирования.

## **С****писок использованных источников**

1. П. Дейтел, *Android* для программистов: создаём приложения, 2013. – 560 с.
2. Колисниченко, Д. Н. Программирование для *Android*. Самоучитель Д. Н. Колисниченко. – СПб.: БХВ-Петербург, 2012. – 272 с.
3. Рето Майер *Android*. Программирование приложений для планшетных компьютеров и смартфонов, 2011– 222 с.
4. Шилдт Герберт. *Java* 8: руководство для начинающих, 6-е изд., 2015. - 720 с.
5. Роберт Седжвик, Кевин Уэйн "Алгоритмы на *Java*", 4-е издание

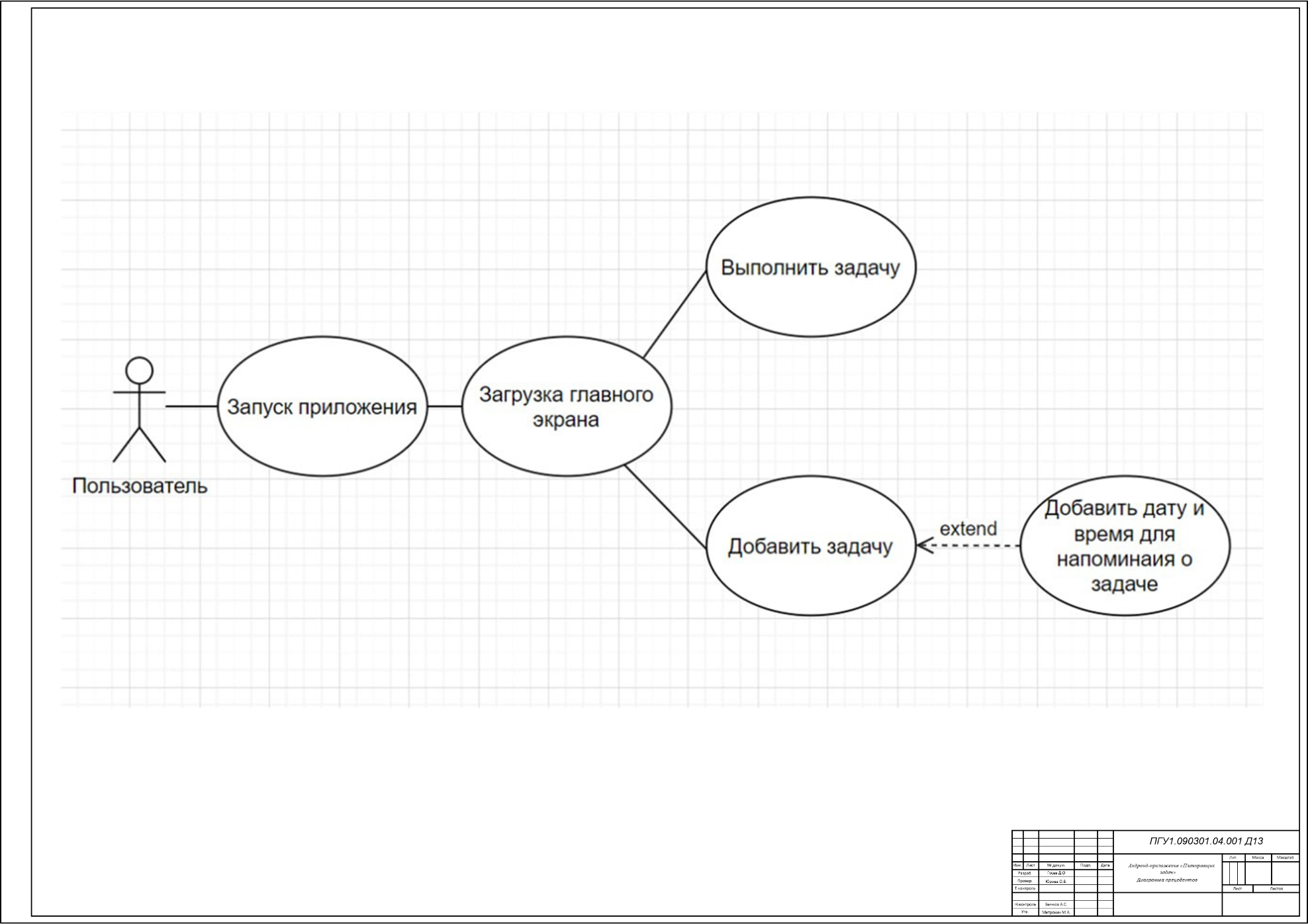
## **Приложение А**

(Обязательное)

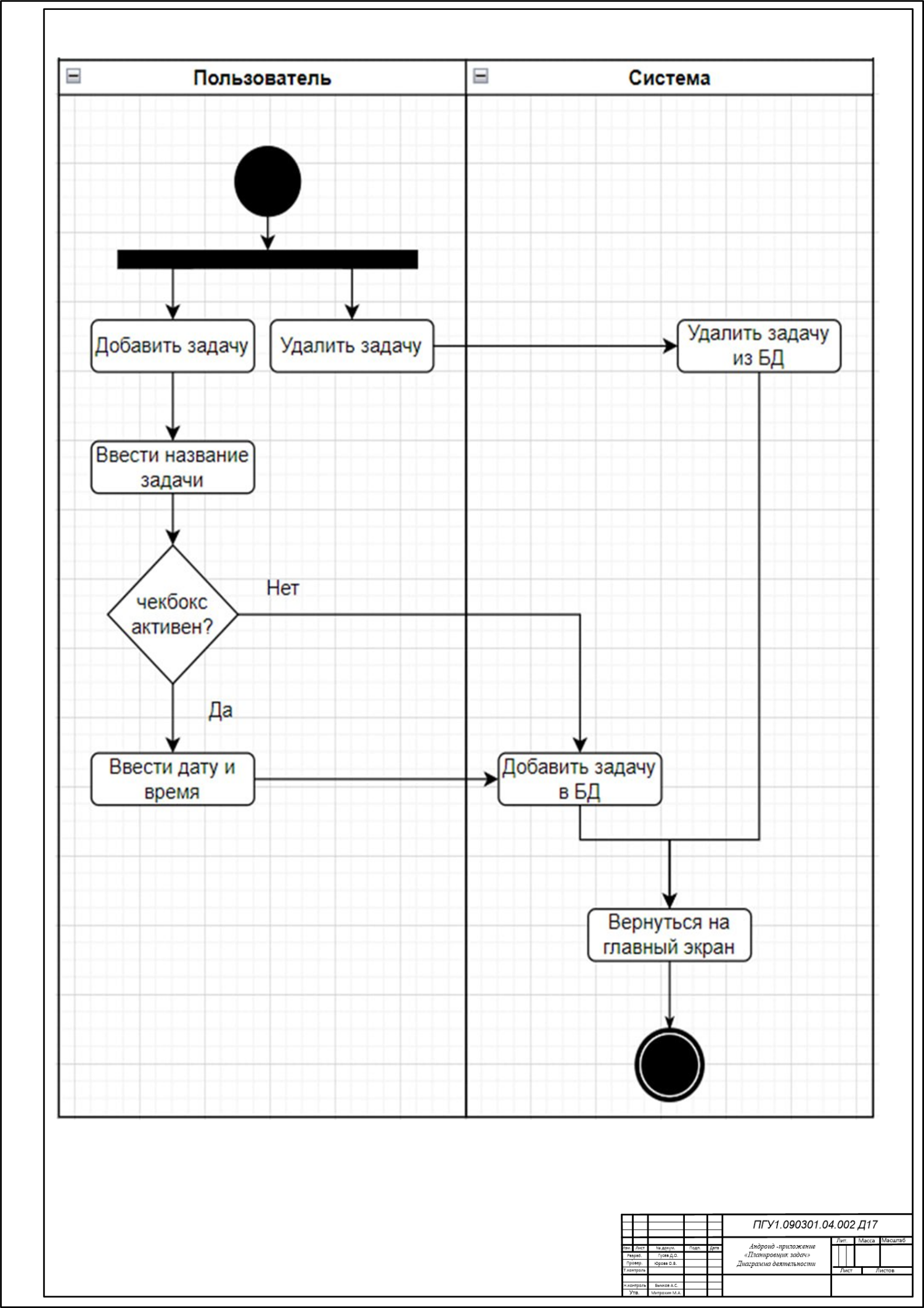
Андроид-приложение «Планировщик задач»

*UML* диаграммы

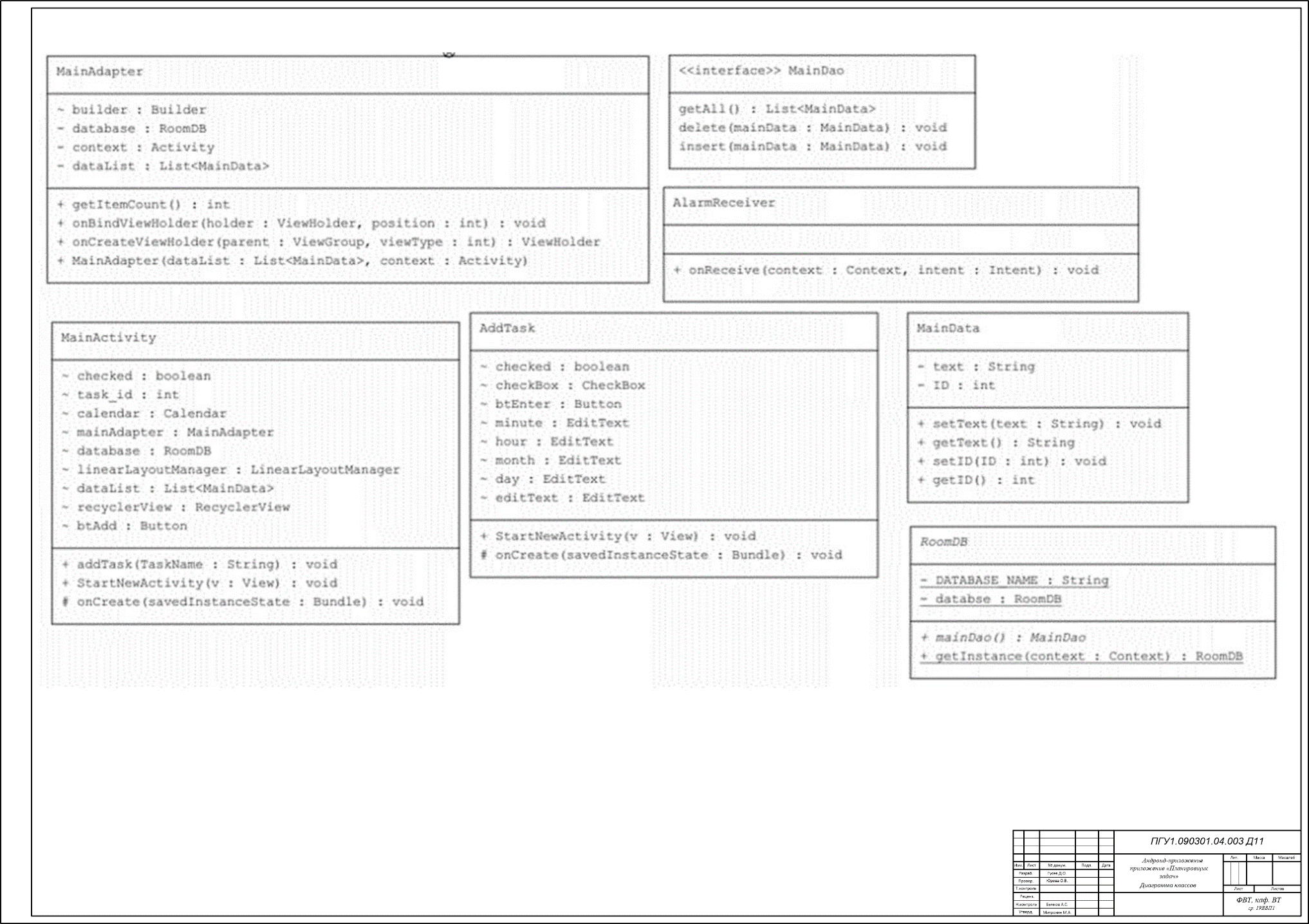
**Приложение А.1 – *UML* – диаграмма прецедентов**

****

**Приложение А.2 –*UML* – диаграмма деятельности**

****

**Приложение А.3 – *UML* – диаграмма классов**



## **Приложение Б**

(Обязательное)

Андроид-приложение «Планировщик задач»

Листинг программы

Файл*AddTask.java*

package com.example.mytodoapplication;  
  
 import android.content.Intent;  
 import android.os.Bundle;  
 import android.view.View;  
 import android.widget.Button;  
 import android.widget.CheckBox;  
 import android.widget.CompoundButton;  
 import android.widget.EditText;  
  
 import androidx.appcompat.app.AppCompatActivity;  
  
  
 public class AddTask extends AppCompatActivity {  
 EditText editText;  
 EditText day;  
 EditText month;  
 EditText hour;  
 EditText minute;  
 Button btEnter;  
 CheckBox checkBox;  
 boolean checked=false;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.add\_activity);  
 editText=findViewById(R.id.edit\_text);  
 day=findViewById(R.id.editTextDay);  
 month=findViewById(R.id.editTextMonth);  
 hour=findViewById(R.id.editTextHour);  
 minute=findViewById(R.id.editTextMinute);  
 btEnter=findViewById(R.id.btn\_enter);  
 checkBox=findViewById(R.id.checkBox);  
  
 day.setEnabled(false);  
 month.setEnabled(false);  
 hour.setEnabled(false);  
 minute.setEnabled(false);  
  
 checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton compoundButton, boolean b) {  
 if (b){  
 day.setEnabled(true);  
 month.setEnabled(true);  
 hour.setEnabled(true);  
 minute.setEnabled(true);

checked=true;  
 }  
 else {  
 day.setEnabled(false);  
 month.setEnabled(false);  
 hour.setEnabled(false);  
 minute.setEnabled(false);  
 checked=false;  
 }  
 }  
 });  
 }  
 public void StartNewActivity(View v){  
 Intent intent = new Intent(this, MainActivity.class);  
 String TaskName = ((EditText)findViewById(R.id.edit\_text)).getText().toString();  
 if(checked){  
 int day = Integer.parseInt(((EditText)findViewById(R.id.editTextDay)).getText().toString());  
 int month = Integer.parseInt(((EditText)findViewById(R.id.editTextMonth)).getText().toString());  
 int hour = Integer.parseInt(((EditText)findViewById(R.id.editTextHour)).getText().toString());  
 int minute = Integer.parseInt(((EditText)findViewById(R.id.editTextMinute)).getText().toString());  
 intent.putExtra("day", day);  
 intent.putExtra("month", month);  
 intent.putExtra("hour", hour);  
 intent.putExtra("minute", minute);  
 }  
 intent.putExtra("task\_name", TaskName);  
 intent.putExtra("haveDate", checked);  
 startActivity(intent);  
 }  
  
}

Файл*AlarmReceiver.java*

package com.example.mytodoapplication;  
  
import android.app.NotificationManager;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
  
import androidx.core.app.NotificationCompat;

public class AlarmReceiver extends BroadcastReceiver {  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 String TaskName = intent.getStringExtra("task\_name");  
 int notificationId = intent.getIntExtra("id", 0);  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(context, "channel\_id")  
 .setSmallIcon(R.drawable.ic\_launcher\_background)  
 .setContentTitle("Напоминание")  
 .setContentText(TaskName)  
 .setPriority(NotificationCompat.PRIORITY\_DEFAULT);  
  
 NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);  
 notificationManager.notify(notificationId, builder.build());  
  
  
 }  
}

Файл*MainActivity.java*

package com.example.mytodoapplication;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
  
import android.app.AlarmManager;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.List;  
  
  
  
public class MainActivity extends AppCompatActivity {  
 Button btAdd;  
 RecyclerView recyclerView;  
  
 List<MainData> dataList=new ArrayList<>();  
 LinearLayoutManager linearLayoutManager;

RoomDB database;  
  
 MainAdapter mainAdapter;  
 Calendar calendar = Calendar.getInstance();  
  
 int task\_id;  
 boolean checked;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 btAdd=findViewById(R.id.bt\_add);  
 recyclerView=findViewById(R.id.recycler\_view);  
  
 database=RoomDB.getInstance(this);  
  
 dataList=database.mainDao().getAll();  
  
 linearLayoutManager =new LinearLayoutManager(this);  
  
 recyclerView.setLayoutManager(linearLayoutManager);  
  
 mainAdapter=new MainAdapter(dataList,MainActivity.this);  
  
 recyclerView.setAdapter(mainAdapter);  
  
  
  
 if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {  
 NotificationChannel channel = new NotificationChannel("channel\_id", "channel\_name", NotificationManager.IMPORTANCE\_DEFAULT);  
 NotificationManager notificationManager = getSystemService(NotificationManager.class);  
 notificationManager.createNotificationChannel(channel);  
 }  
  
 Bundle arguments = getIntent().getExtras();  
 String TaskName;  
 int day, month, hour, minute;  
 if (arguments != null) {  
 TaskName = arguments.get("task\_name").toString();  
 checked = arguments.getBoolean("haveDate");  
 if (checked){  
 day = arguments.getInt("day");  
 month = arguments.getInt("month");  
 hour = arguments.getInt("hour");  
 minute = arguments.getInt("minute");  
  
 calendar.set(Calendar.DAY\_OF\_MONTH, day);  
 calendar.set(Calendar.MONTH, month - 1);  
 calendar.set(Calendar.HOUR\_OF\_DAY, hour);  
 calendar.set(Calendar.MINUTE, minute);  
  
 getIntent().removeExtra("day");  
 getIntent().removeExtra("month");  
 getIntent().removeExtra("hour");  
 getIntent().removeExtra("minute");  
  
 AlarmManager alarmManager = (AlarmManager) this.getSystemService(Context.ALARM\_SERVICE);  
 Intent intent = new Intent(this, AlarmReceiver.class);  
 intent.putExtra("task\_name", TaskName);  
 intent.putExtra("id", task\_id);  
 PendingIntent pendingIntent = PendingIntent.getBroadcast(this, task\_id, intent, 0);  
  
 if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.KITKAT) {  
 alarmManager.setExact(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), pendingIntent);  
 }  
 }  
 addTask(TaskName);  
 getIntent().removeExtra("task\_name");  
  
 }  
  
 mainAdapter.notifyDataSetChanged();  
  
 };  
  
 public void StartNewActivity(View v){  
 Intent add\_intent = new Intent(this, AddTask.class);  
 startActivity(add\_intent);  
 }  
 public void addTask(String TaskName){  
  
 MainData data=new MainData();  
  
 data.setText(TaskName);  
  
  
 database.mainDao().insert(data);  
  
 dataList.clear();  
 Toast.makeText(MainActivity.this,"Задача добавлена!",Toast.LENGTH\_LONG).show();  
 dataList.addAll(database.mainDao().getAll());  
 data = dataList.get(dataList.size()-1);  
 task\_id = data.getID();  
  
 }  
  
  
  
}

Файл*MainAdapter.java*

package com.example.mytodoapplication;  
import android.app.Activity;  
import android.app.AlarmManager;  
import android.app.AlertDialog;  
import android.app.PendingIntent;  
import android.content.DialogInterface;  
import android.content.Intent;  
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 java.util.List;  
  
public class MainAdapter extends RecyclerView.Adapter<MainAdapter.ViewHolder> {  
  
 private List<MainData> dataList;  
 private Activity context;  
 private RoomDB database;  
  
 AlertDialog.Builder builder;  
  
  
  
 public MainAdapter(List<MainData> dataList, Activity context) {  
 this.dataList = dataList;  
 this.context = context;  
 notifyDataSetChanged();  
 }  
  
 @NonNull  
  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view= LayoutInflater.from(parent.getContext())  
 .inflate(R.layout.list\_row\_main,parent,false);  
  
  
 return new ViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull MainAdapter.ViewHolder holder, int position) {

MainData data=dataList.get(position);

database=RoomDB.getInstance(context);  
 holder.textView.setText(data.getText());  
  
  
 holder.btCheck.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 builder= new AlertDialog.Builder(v.getContext());  
  
 builder.setMessage("Подтвердить выполнение задачи?")  
 .setCancelable(false)  
 .setPositiveButton("Да", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
  
 MainData d=dataList.get(holder.getAdapterPosition());  
  
 database.mainDao().delete(d);  
 int position = holder.getAdapterPosition();  
 dataList.remove(position);  
 notifyItemRemoved(position);  
 notifyItemRangeChanged(position,dataList.size());  
  
 System.out.println(d.getID());  
 Intent intent = new Intent(context, AlarmReceiver.class);  
  
 PendingIntent pendingIntent = PendingIntent.getBroadcast(context, d.getID(), intent, 0);  
  
 AlarmManager alarmManager = null;  
 if (android.os.Build.VERSION.SDK\_INT >= android.os.Build.VERSION\_CODES.M) {  
 alarmManager = (AlarmManager) context.getSystemService(context.ALARM\_SERVICE);  
 }  
  
 alarmManager.cancel(pendingIntent);  
  
 Toast.makeText(context,"Задача выполнена!",Toast.LENGTH\_LONG).show();  
 }  
 })  
 .setNegativeButton("Нет", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
 dialog.cancel();  
 }

});  
 AlertDialog alert = builder.create();  
 alert.setTitle("Подтверждение");  
 alert.show();  
  
 }  
 });  
  
 }  
  
 @Override  
 public int getItemCount() {  
 return dataList.size();  
 }  
  
 public class ViewHolder extends RecyclerView.ViewHolder {  
  
 TextView textView;  
 ImageView btCheck;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 textView=itemView.findViewById(R.id.text\_view);  
 btCheck=itemView.findViewById(R.id.bt\_check);  
 }  
 }  
}

Файл*MainDao.java*

package com.example.mytodoapplication;  
import androidx.room.Dao;  
import androidx.room.Delete;  
import androidx.room.Insert;  
import androidx.room.Query;  
  
import java.util.List;  
  
import static androidx.room.OnConflictStrategy.REPLACE;  
  
@Dao  
public interface MainDao {  
  
 @Insert(onConflict = REPLACE)  
 void insert(MainData mainData);  
  
 @Delete  
 void delete(MainData mainData);  
  
 @Query("SELECT \* FROM table\_name")  
 List<MainData> getAll();  
}

Файл*MainDao.java*

package com.example.mytodoapplication;  
import androidx.room.ColumnInfo;  
import androidx.room.Entity;  
import androidx.room.PrimaryKey;  
  
import java.io.Serializable;  
  
@Entity(tableName = "table\_name")  
public class MainData implements Serializable {  
 @PrimaryKey(autoGenerate = true)  
  
 private int ID;  
  
  
 @ColumnInfo(name = "text")  
  
 private String text;  
  
  
  
 public int getID() {  
 return ID;  
 }  
  
 public void setID(int ID) {  
 this.ID = ID;  
 }  
  
 public String getText() {  
 return text;  
 }  
  
 public void setText(String text) {  
 this.text = text;  
 }  
  
  
}

Файл*RoomDB.java*

package com.example.mytodoapplication;  
import androidx.room.Database;  
import androidx.room.Room;  
import androidx.room.RoomDatabase;  
import android.content.Context;  
  
@Database(entities = {MainData.class},version = 1,exportSchema = false)  
  
public abstract class RoomDB extends RoomDatabase {

private static RoomDB databse;

private static String DATABASE\_NAME="database";  
  
 public synchronized static RoomDB getInstance(Context context){  
 if(databse==null){  
 databse= Room.databaseBuilder(context.getApplicationContext(),RoomDB.class,DATABASE\_NAME)  
 .allowMainThreadQueries()  
 .fallbackToDestructiveMigration()  
 .build();  
 }  
  
 //return db  
 return databse;  
 }  
  
  
 public abstract MainDao mainDao();  
  
}

Файл*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="8dp">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:id="@+id/bt\_add"  
 android:onClick="StartNewActivity"  
 android:text="Новая задача"/>  
  
  
 </LinearLayout>  
  
 <androidx.recyclerview.widget.RecyclerView

android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/recycler\_view"  
 android:layout\_marginTop="8dp"

tools:listitem="@layout/list\_row\_main"/>  
  
</LinearLayout>

Файл*add\_activity.xml*

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <EditText  
 android:id="@+id/edit\_text"  
 android:layout\_width="366dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="148dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:minHeight="48dp"  
 android:text=""  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="SpeakableTextPresentCheck" />  
  
 <Button  
 android:id="@+id/btn\_enter"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="StartNewActivity"  
 android:text="Добавить"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.503"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.838" />  
  
 <EditText  
 android:id="@+id/editTextDay"  
 android:layout\_width="48dp"  
 android:layout\_height="48dp"  
 android:ems="10"  
 android:inputType="number"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.303"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.535" />

<EditText  
 android:id="@+id/editTextMinute"  
 android:layout\_width="48dp"  
 android:layout\_height="48dp"  
 android:ems="10"  
 android:inputType="number"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.6"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.673" />  
  
 <TextView  
 android:id="@+id/textView4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Время"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.07"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.665" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Месяц"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.709"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.529" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="День"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.067"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.529" />  
  
 <EditText

android:id="@+id/editTextMonth"  
 android:layout\_width="48dp"  
 android:layout\_height="48dp"  
 android:ems="10"  
 android:inputType="number"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.942"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.535" />  
  
 <EditText  
 android:id="@+id/editTextHour"  
 android:layout\_width="48dp"  
 android:layout\_height="48dp"  
 android:ems="10"  
 android:inputType="number"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.363"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.673" />  
  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text=":"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.491"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.665" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="42dp"  
 android:text="Введите задачу:"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/edit\_text"

app:layout\_constraintStart\_toStartOf="@+id/edit\_text"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.873"  
 tools:ignore="MissingConstraints" />  
  
 <CheckBox  
 android:id="@+id/checkBox"

android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Напомнить"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.095"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.377" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Файл*list\_row\_main.xml*

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="4dp">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/text\_view"  
 android:layout\_width="339dp"  
 android:layout\_height="32dp"  
 android:layout\_weight="1"  
 android:textColor="@color/black"  
 android:textSize="14sp" />  
  
 <ImageView  
 android:id="@+id/bt\_check"  
 android:layout\_width="32dp"  
 android:layout\_height="32dp"  
  
 app:srcCompat="@drawable/ic\_check" />  
  
  
 </LinearLayout>  
</androidx.cardview.widget.CardView>

Файл*AndroidManifest.xml*

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.mytodoapplication">  
 <uses-permission android:name="android.permission.SET\_ALARM" />  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.VIBRATE" />

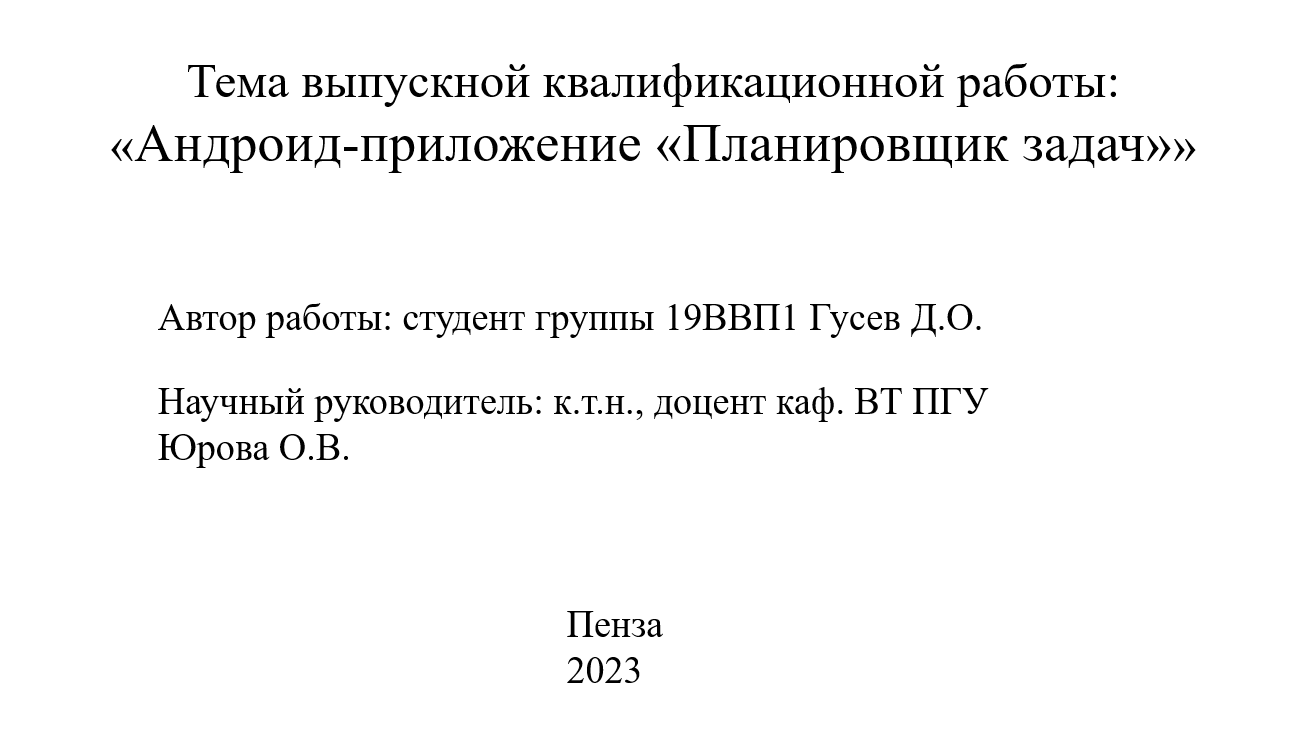
<uses-permission android:name="android.permission.ACCESS\_NOTIFICATION\_POLICY" />  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
 <uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />  
 <uses-permission android:name="android.permission.WAKE\_LOCK" />  
 <uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyTodoApplication">  
 <activity android:name="com.example.mytodoapplication.MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".AddTask">  
  
 </activity>  
 <receiver android:name=".AlarmReceiver" />  
 </application>  
  
</manifest>

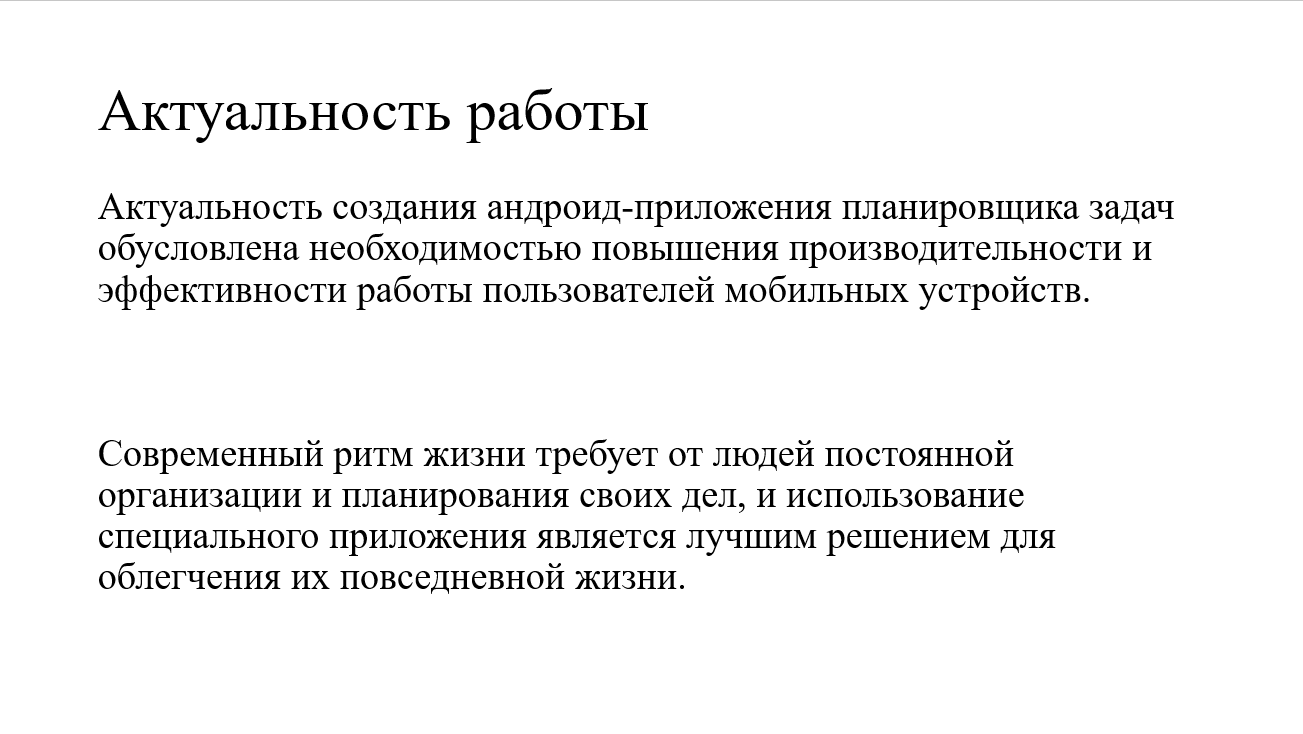
## **Приложение В**

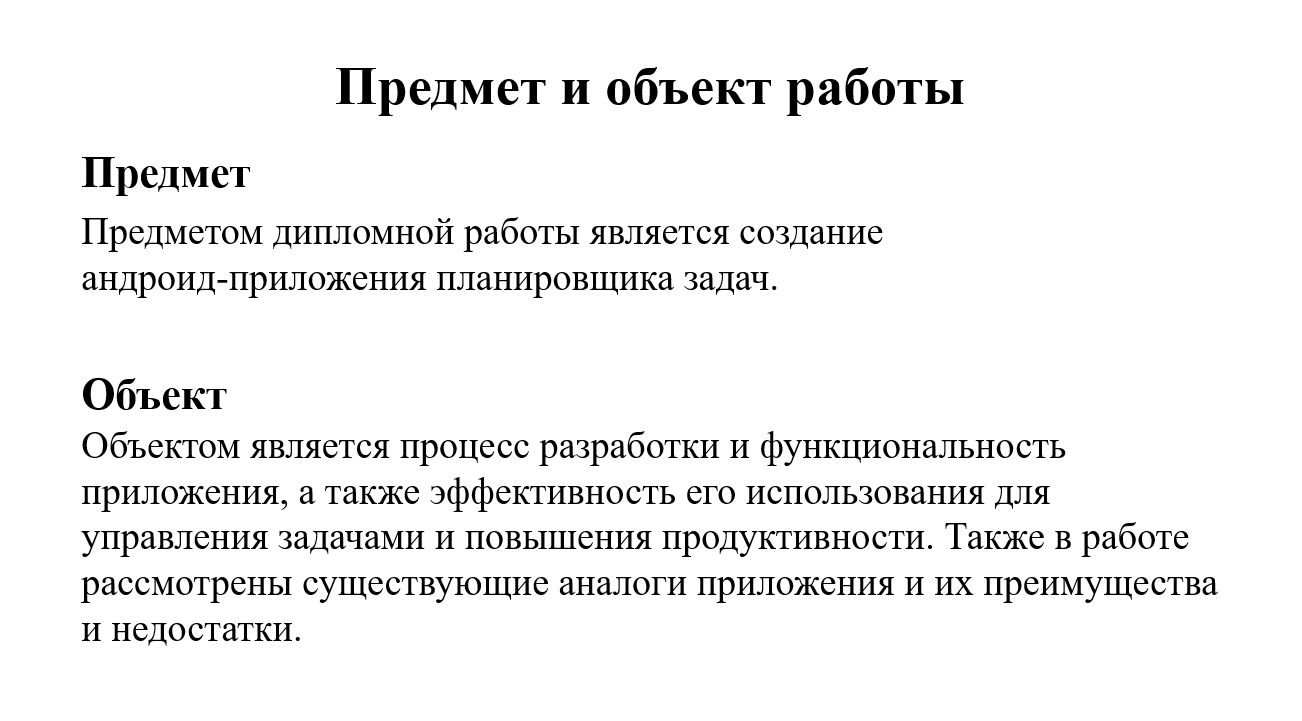
(Обязательное)

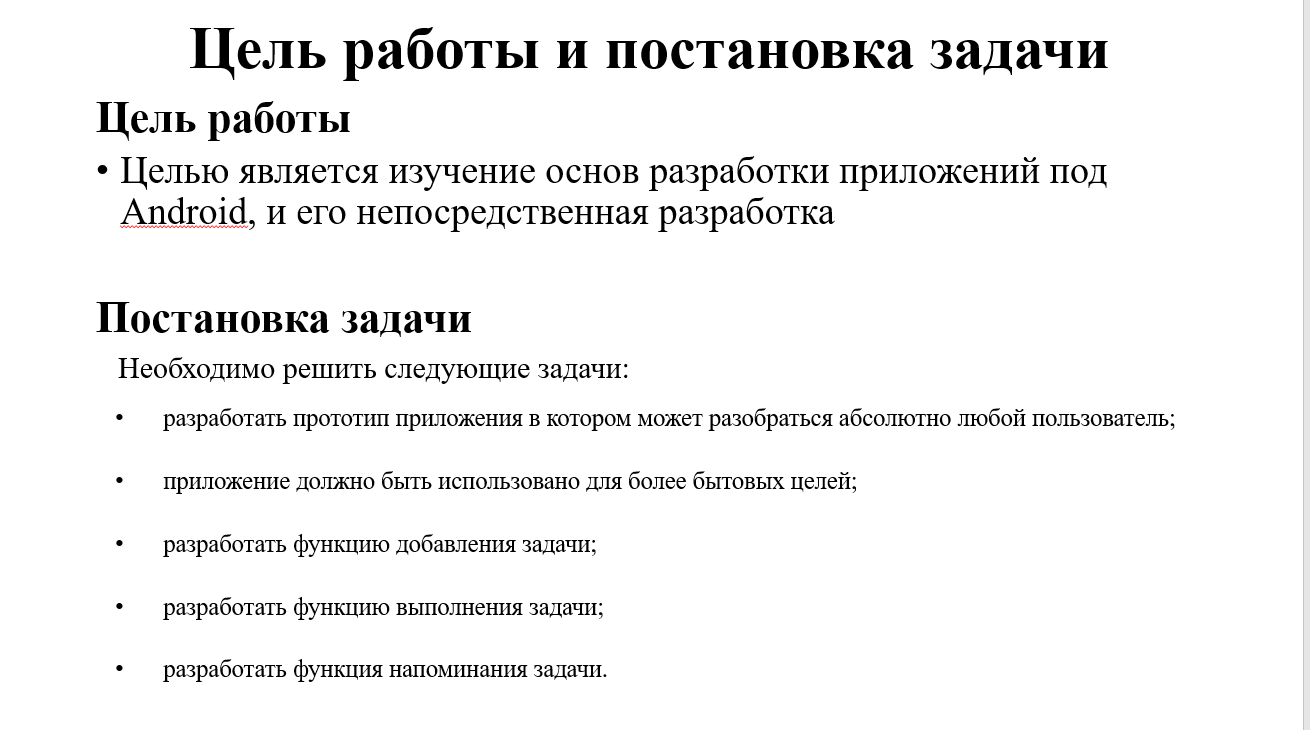
Андроид-приложение «Планировщик задач»

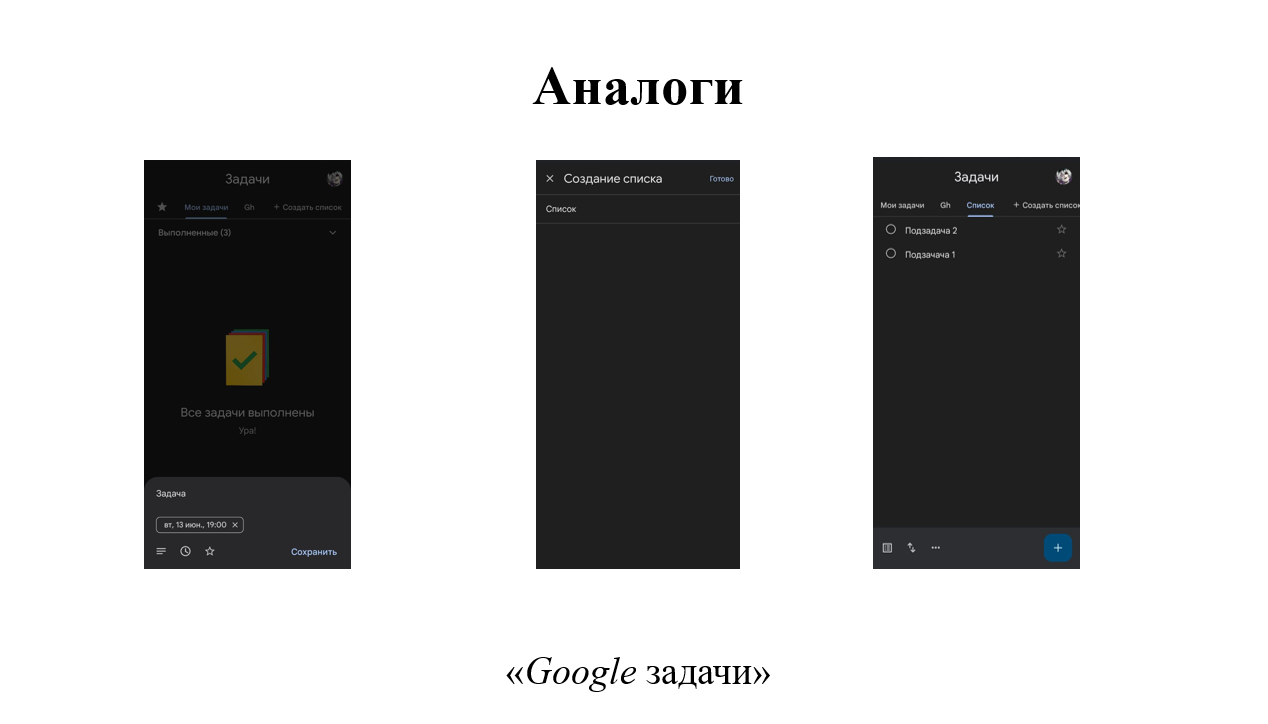
Презентация

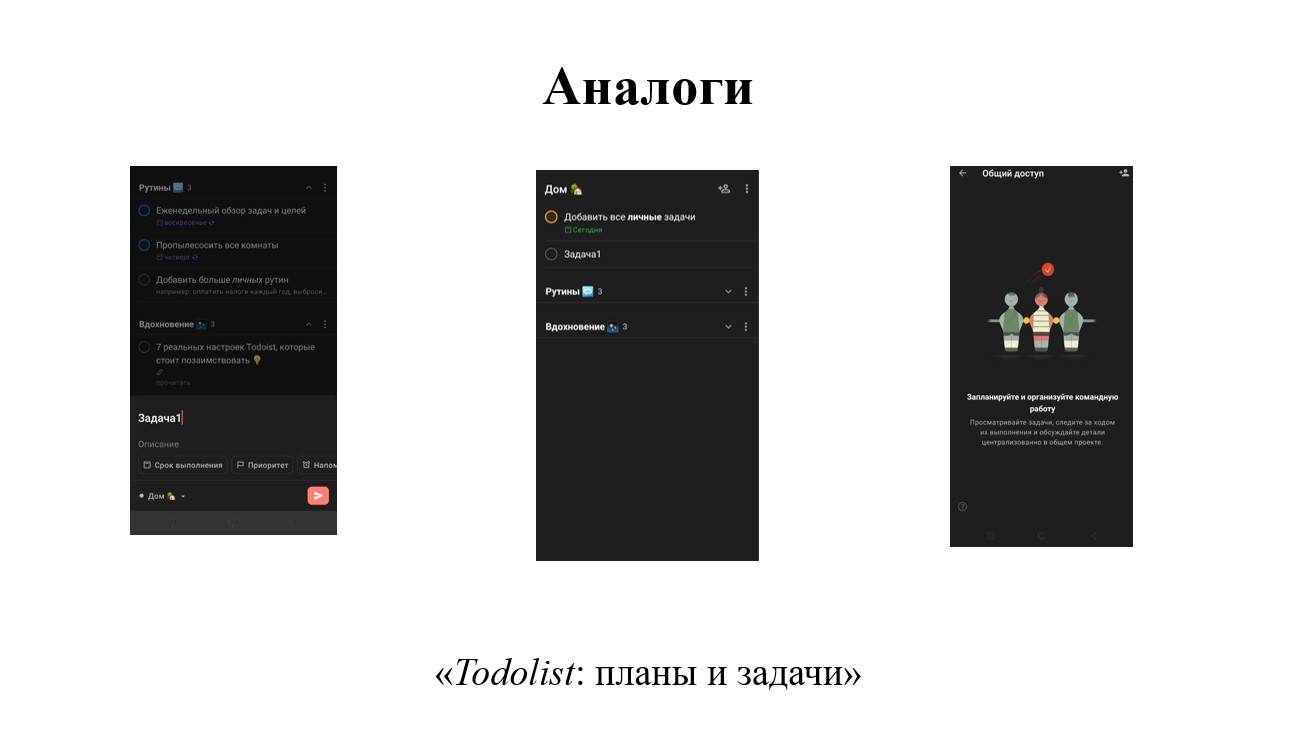


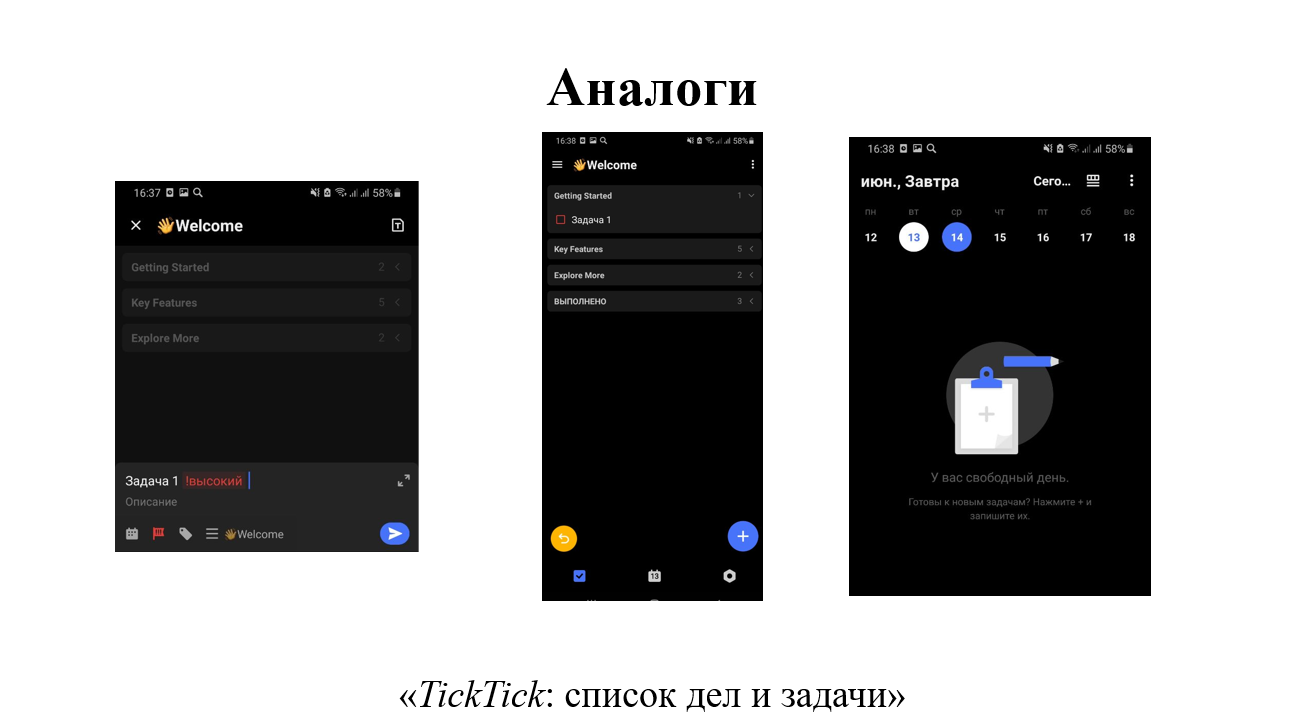


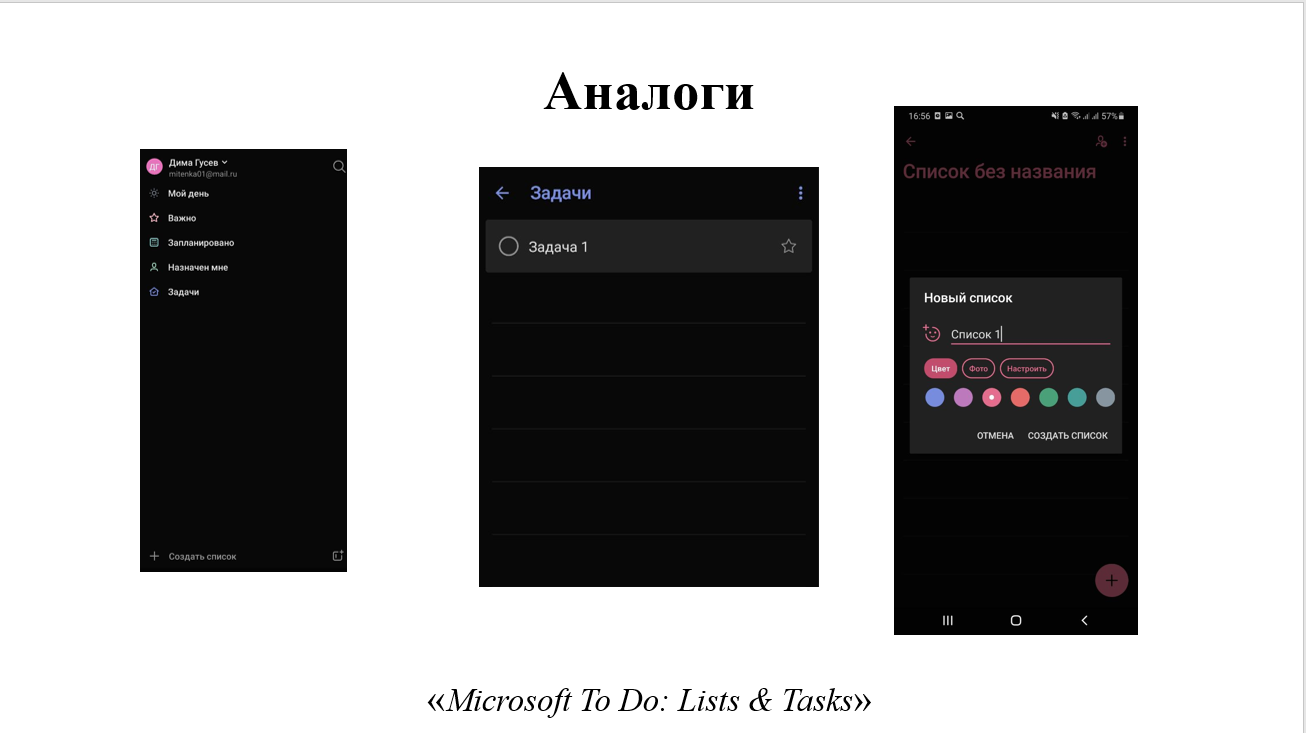




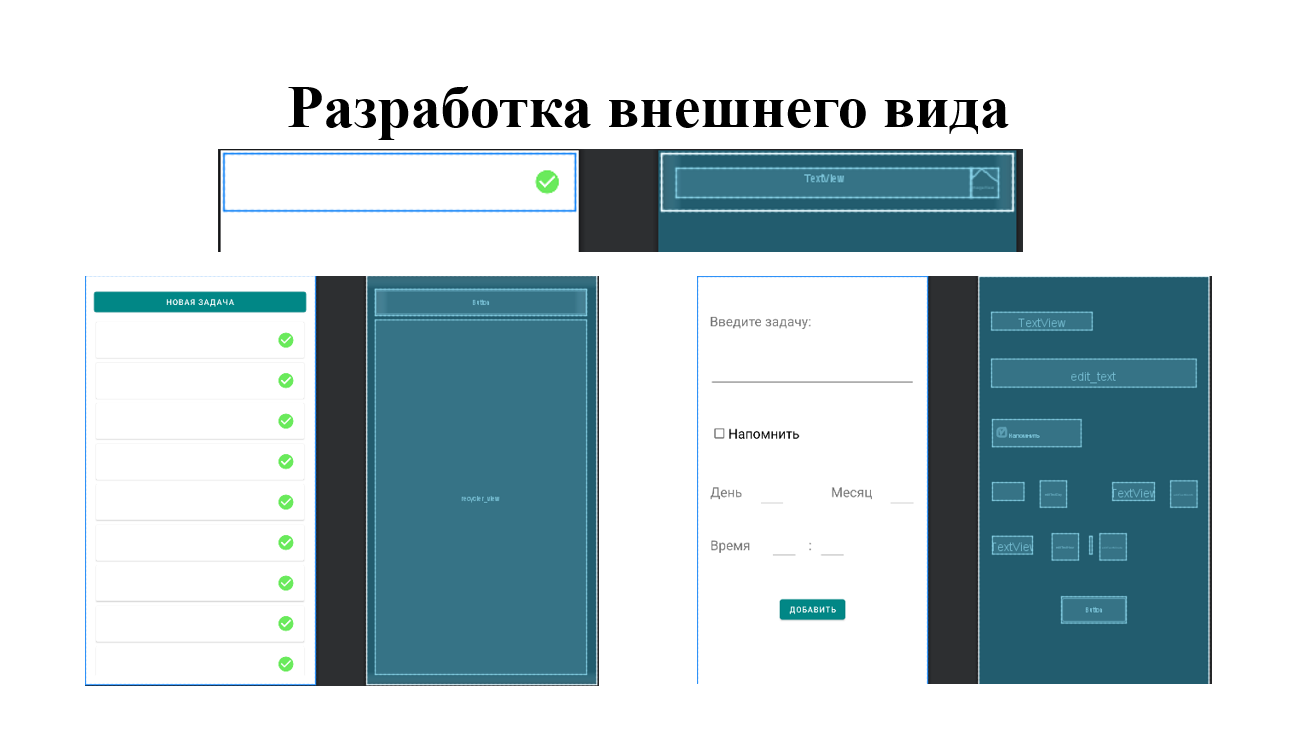








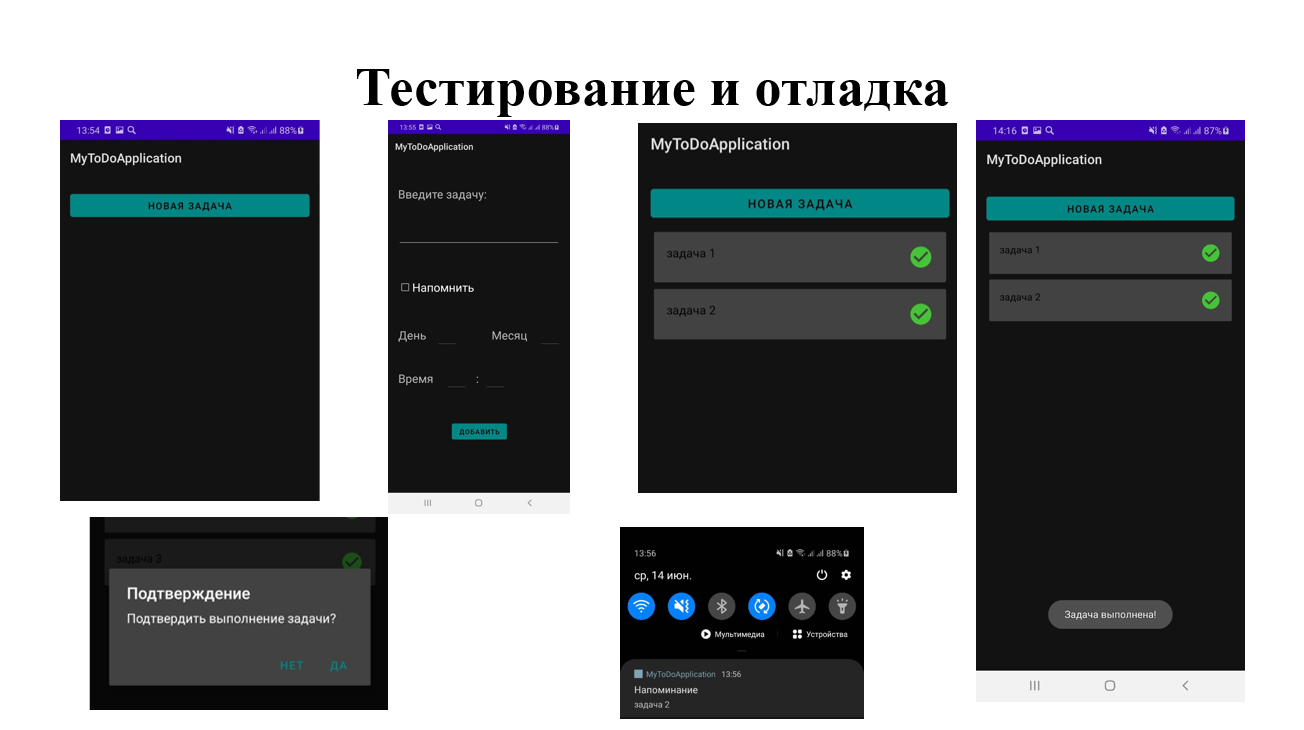


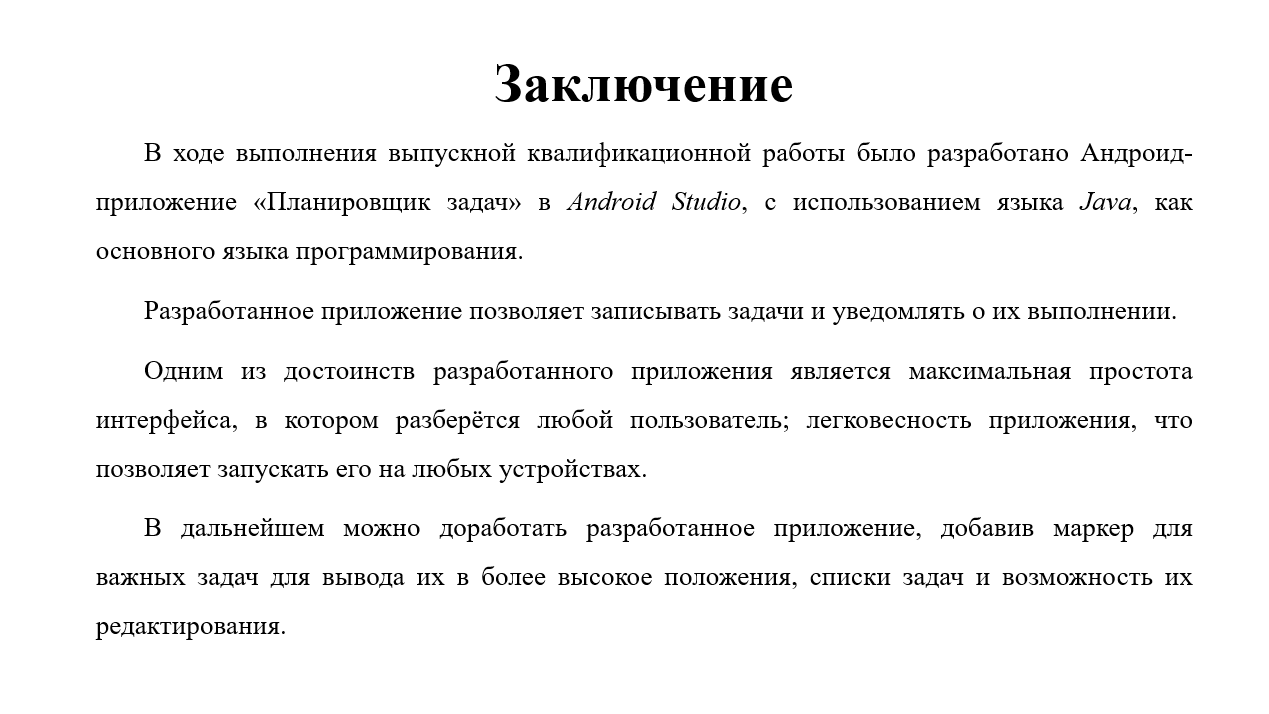


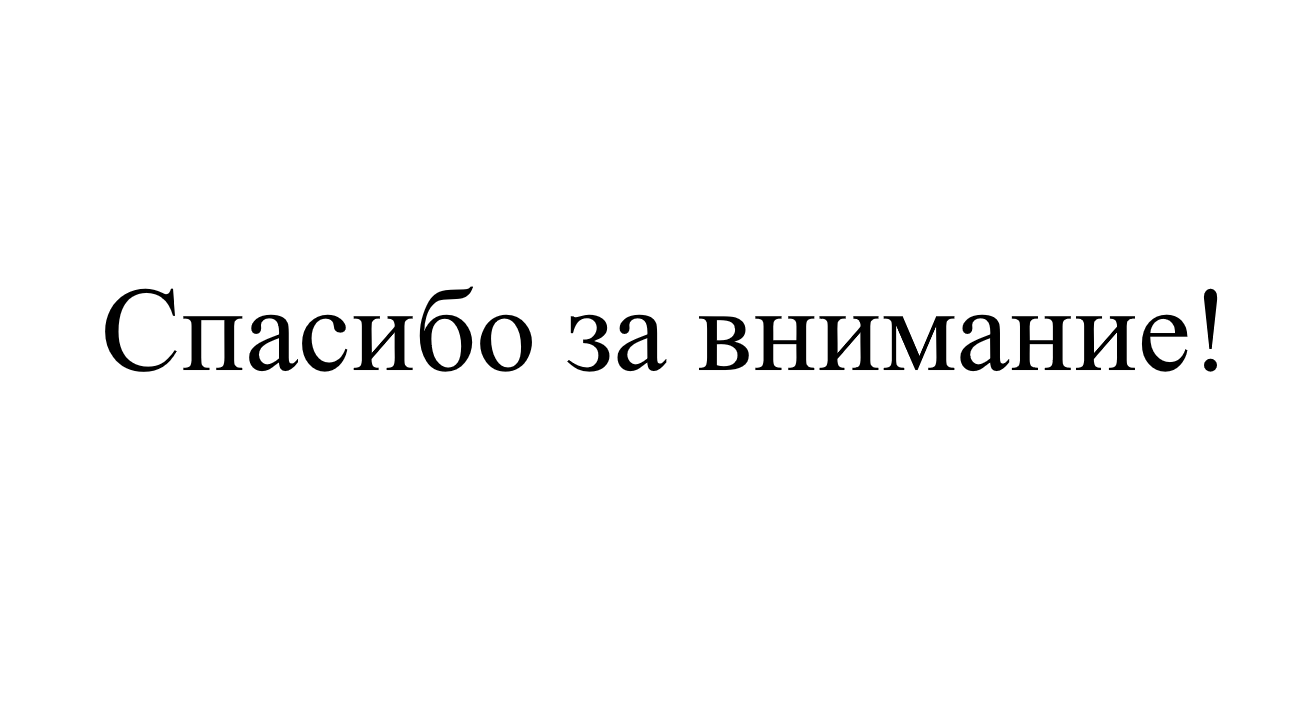












Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Пензенский государственный университет»

|  |  |  |
| --- | --- | --- |
| *Факультет* |  | *Кафедра* |
| вычислительной техники |  | «Вычислительная техника» |

«Утверждаю»

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_М.А. Митрохин

« 26 »\_\_\_\_апреля\_\_\_\_20\_23\_г.

**ЗАДАНИЕ**

**НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Гусев Дмитрий Олегович гр 19ВВП1 факультета вычислительной техники.

2.Тема работы Образовательное приложение «Магия знаний»

Тема утверждена приказом ПГУ № 458/о от "26" апреля 2023 г.

3. Руководитель бакалаврской работы к.т.н., доцент, каф. ВТ ПГУ Юрова О.В.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать андроид-приложение Планировщик задач. Программа должна добавлять, отображать и выполнять задачи; упоминать об их выполнении.

5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметных областей: изучить существующие образовательные приложения, выделить их достоинства и недостатки. Выбрать язык программирования, который подойдет для написания логики приложения. Определить архитектуру приложения и вспомогательные средства разработки.

6. Календарный график выполнения работы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование этапов работы | Объем работы | Срок выполнения | Подпись руководителя |
| 1. | Изучение предметной области | 50 | 26.04.2023- 30.04.2023 |  |
| 2. | Анализ требований | 10 | 01.05.2023- 02.05.2023 |  |
| 3. | Выбор технологий, применяемых при проектировании и разработке | 50 | 03.05.2023- 07.05.2023 |  |
| 4. | Описание алгоритмов | 40 | 08.05.2023- 11.05.2023 |  |
| 5. | Разработка приложения «Андроид-приложение «Планировщик задач»» | 150 | 12.05.2023- 29.05.2023 |  |
| 6. | Оформление пояснительной записки | 24 | 30.05.2023- 31.05.2023 |  |

Дата выдачи задания "26" апреля 2023 г.

Руководитель бакалаврской работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Юрова О.В.

*(подпись, дата)*

Задание к исполнению принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гусев Д.О.

*(подпись, дата)*

Работу к защите допустить

Декан факультета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фионова Л.Р.

*(подпись, дата)*