

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Информатика и вычислительная техника»

БАКАЛАВРСКАЯ РАБОТА

на тему

ПРИЛОЖЕНИЕ ДЛЯ УДАЛЕННОГО
УПРАВЛЕНИЯ ПК С МОБИЛЬНОГО
УСТРОЙСТВА

Студент

М.А. 30.05.25 Мубаряков Азамат Артурович
(подпись, дата) (ФИО полностью)

Руководитель

Ю.В. 30.05.25 Юрова О.В.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр

А.С. 30.05.25 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 28.05.2025 г. № 16)

Заведующий кафедрой

М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от

16 июня 2025 г. № 38)

Секретарь ГЭК

Е.И. Гурин
(подпись) (фамилия, инициалы)

Пенза 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 06 » 05 20 25 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Мубаряков Азамат Артурович гр. 21ВВП1 факультета вычислительной техники
2. Тема работы. Приложение для удаленного управления ПК с мобильного устройства.

Тема утверждена приказом ПГУ № 601/о от "06" мая 2025 г.

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

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

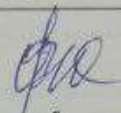
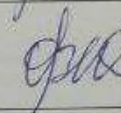
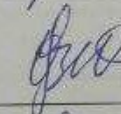
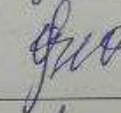
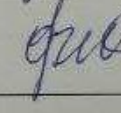
Разработать настольный и мобильный клиент приложения "CommanderTask", позволяющее дистанционно управлять устройствами в сети. Язык программирования Java, среда разработки AndroidStudio и IntelliJ IDEA. Программа должна обладать интуитивно понятным интерфейсом и использовать следующие технологии:

1. SharedPreferences
2. Private Config Files
3. Асинхронное программирование
4. Material design 3

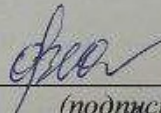
5. Перечень подлежащих разработке вопросов

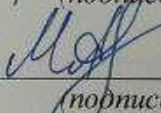
Требуется произвести анализ предметной области: изучить существующие аналоги, изучить способы проектирования и разработки мобильного приложения. Выбрать язык программирования, среду разработки и базу данных.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	45	06.05.2025 – 08.05.2025	
2	Анализ требований	35	09.05.2025 – 13.05.2025	
3	Выбор технологий, применяемых при проектировании и разработке	55	14.05.2025 – 18.05.2025	
4	Создание графического интерфейса	65	19.05.2025 – 22.05.2025	
5	Разработка мобильного приложения	84	23.05.2025 – 26.05.2025	
6	Оформление пояснительной записки	40	27.05.2025 – 30.05.2025	

Дата выдачи задания "06" мая 2025 г.

Руководитель бакалаврской работы  06.05.25 Юрова О.В.
(подпись, дата)

Задание к исполнению принял студент  06.05.25 Мубаряков А.А.
(подпись, дата)

Работу к защите допустить
Декан факультета  16.06.2025 Фионова Л.Р.
(подпись, дата)

Реферат

Бакалаврская работа содержит 99 листов, 30 рисунков, 2 таблицы, 8 источника, 3 приложения.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, НАСТОЛЬНОЕ, ПРИЛОЖЕНИЕ, ТЕСТИРОВАНИЕ, SHARED PREFERENCE, PRIVATE CONFIG FILES, ANDROID STUDIO, JAVA, ANDROID

Цель работы – разработать мобильное приложение, осуществляющее удаленное управление ПК по сети.

Мобильное приложение должно позволять пользователем просматривать список доступных к подключению устройств, выполнять безопасное подключение к ним, выполнять базовые функции по управлению ПК. Помимо отправки удаленных команд, необходимо реализовать управлением курсором и эмуляцию нажатия системных клавиш.

В конечном итоге было разработано настольное приложение для Windows и мобильное приложение для Android на языке Java, которые удовлетворяют всем поставленным задачам. Была реализована регистрация и сохранение данных подключаемых устройств с помощью SharedPreferences и Private Config Flies. Само приложение работает на устройствах ОС Android и Windows и разрабатывалось в специальной среде разработки Android Studio и IntelliJ IDEA.

					ПГУ1.090301.16.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Приложение для удаленного управления ПК с мобильного устройства	Лит.	Лист	Листов	
Разраб.		Мубарянов А.А.		20.05.25			2	99	
Провер.		Юрова О.В.		20.05.25					
Н. Контр		Бычков А.С.		20.05.25					
Утверд.		Митрохин М.А.		20.05.25					
					ФВТ, каф. ВТ гр. 21ВВП1				

Содержание

Перечень сокращений и обозначений.....	5
Введение	6
1 Приложение «CommanderTask» для ОС Android и Windows	8
1.1 Анализ темы.....	8
1.2 Анализ аналогов	11
1.2.1 Мобильное приложение «PC Remote»	11
1.2.2 Мобильное приложение «AnyDesk»	12
1.2.3 Мобильное приложение «Связь с Windows»	15
1.2.4 Мобильное приложение «Unified Remote».....	16
1.2.5 Вывод	18
1.3 Постановка задачи.....	19
2 Выбор инструментов разработки	20
2.1 Выбор языка программирования	20
2.1.1 Язык программирования «Java»	20
2.1.2 Язык программирования «Kotlin»	20
2.1.3 Язык программирования «C++»	21
2.1.4 Вывод	21
2.2 Выбор среды разработки	21
2.3 Выбор базы данных.....	22
3 Архитектура приложения	23
3.1 Архитектурные решения в проекте CommanderTask.....	23
3.1.1 Принципы организации кода	23
3.2 Структурные компоненты	23
3.2.1 Управление конфигурацией (Data Layer)	23
3.2.2 Сетевое взаимодействие (Service Layer).....	23
3.2.3 Презентационный слой (UI Components).....	24
3.3 Ключевые архитектурные решения.....	24
3.3.1 Разделение UI и бизнес-логики	24
3.3.2 Паттерн "Адаптер"	24

3.3.3 Управление зависимостями	25
3.4 Возможности для развития архитектуры.....	25
3.4.1 Потенциальные улучшения.....	25
3.5 Преимущества выбранного подхода	25
4 Реализация приложения	26
4.1 Описание функционала приложения CommanderTask	26
4.2 Описание работы приложения	30
4.3 Основная реализация приложения	32
4.3.1 UI-слой (Presentation).....	32
4.3.2 Бизнес-логика (Domain).....	33
4.3.3 Слой данных (Data).....	33
4.4 Ключевые отличия от Clean Architecture	33
4.4.1 Смешение слоёв:	33
4.4.2 Управление зависимостями:	33
4.4.3 Работа с данными:.....	33
Пример потока данных по сценарию: Изменение громкости	33
4.5 Преимущества выбранного подхода	34
5 Тестирование.....	35
5.1 Взаимодействие с главным экраном	35
5.2 Отправка команд	39
5.3 Изменение настроек	43
5.4 Вывод.....	45
6 Безопасность жизнедеятельности	46
6.1 Правила и техника безопасности при работе с ПК.	46
6.2 Правила и техника безопасности при работе со смартфоном.....	47
Заключение.....	49
Список использованных источников.....	50
Приложение А UML–диаграммы.....	51
Приложение Б Листинг программы.....	57
Приложение В Презентация	93

Перечень сокращений и обозначений

В данной работе применяются следующие сокращения и обозначения.

ЛКМ – левая кнопка мыши

ОС – операционная система

ПК – персональный компьютер

ПКМ – правая кнопка мыши

ПО – программное обеспечение

CMD – (от англ. Command) – встроенное приложение операционной системы семейства Windows

IDE – (от англ. Integrated Development Environment) – интегрированная среда разработки

IP – (от англ. Internet Protocol) – протокол взаимодействия устройств в сети интернет

TCP – (от англ. Transmission Control Protocol) – протокол передачи данных

UI – (от англ. User Interface) – пользовательский интерфейс

UML – (от англ. Unified Modeling Language) – унифицированный язык моделирования

					<i>ПГУ1.090301.16.001 ПЗ</i>	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

Введение

В современном мире всё больше процессов автоматизируется, а устройства становятся взаимосвязанными. Пользователи стремятся к удобству, особенно когда речь идёт об управлении компьютером без физического доступа к нему. Это особенно актуально для людей, работающих удалённо, а также для тех, кто хочет быстро взаимодействовать с домашним ПК, находясь в другой комнате или вне дома. Одним из решений может стать мобильное приложение для удалённого управления персональным компьютером с телефона.

Для удобства пользователей в качестве управляющего устройства используется мобильное приложение, поскольку смартфон всегда находится под рукой. Такое приложение позволяет осуществлять базовое и расширенное управление ПК: управление медиа (воспроизведение, пауза, регулировка громкости), ввод с клавиатуры и мыши, запуск программ и другие действия.

Мобильное приложение разрабатывается под операционную систему Android с использованием среды разработки Android Studio и языка Java. Серверная часть, устанавливаемая на ПК с Windows, создаётся с помощью IntelliJ IDEA также на языке Java. Обмен данными между клиентом и сервером осуществляется через локальную сеть или интернет с использованием безопасных и надёжных протоколов передачи данных.

Цели выпускной квалификационной работы:

- исследовать актуальность и возможности использования мобильного управления ПК;
- определить функциональность и пользовательский интерфейс приложения;
- разработать современное кроссплатформенное программное обеспечение для удалённого взаимодействия с ПК;
- реализовать возможности управления медиа, ввода с клавиатуры и мыши с мобильного устройства;

- провести тестирование программного обеспечения на различных устройствах и сетевых условиях;
- оценить перспективы внедрения такого приложения в частном и корпоративном использовании;
- предложить направления для дальнейшего расширения функционала (например, удалённая передача файлов, управление питанием ПК и др.);
- проанализировать полученные результаты и сделать выводы о потенциале дальнейшего развития проекта.

					ПГУ1.090301.16.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1 Приложение «CommanderTask» для ОС Android и Windows

1.1 Анализ темы

В наше время пользователи всё активнее стремятся к комфорту, автоматизации и удалённому доступу к своим цифровым устройствам. Одной из востребованных задач является возможность управления персональным компьютером (ПК) с мобильного устройства. Это особенно актуально в ситуациях, когда требуется быстрое вмешательство — например, выключить ПК, управлять воспроизведением медиафайлов или эмулировать нажатия клавиш, находясь вне физического доступа к устройству.

Вот несколько причин, почему использование такого рода решений полезно:

- удалённый доступ и контроль. Приложение позволяет управлять ПК с любой точки, где есть интернет-соединение или локальная сеть, без необходимости находиться рядом с компьютером;

- гибкость и удобство. Пользователь может запускать программы, управлять медиаплеером, выполнять симуляцию ввода с клавиатуры и мыши прямо со смартфона;

- экономия времени. Удалённое управление избавляет от необходимости физически перемещаться к устройству для выполнения базовых действий;

- поддержка автоматизации. Можно интегрировать управление в ежедневные процессы: запуск фильмов, воспроизведение музыки, выключение ПК по расписанию и т.п.

Однако, несмотря на растущий интерес, многие пользователи откладывают установку и настройку таких решений, либо сталкиваются с техническими трудностями при выборе подходящего программного обеспечения. Поэтому важно создать понятное, функциональное и безопасное приложение, которое поможет пользователю быстро организовать удалённое управление своим ПК.

С точки зрения цифрового взаимодействия, формирование привычки использовать удалённый доступ также требует времени. Чтобы регулярно применять приложение и полностью интегрировать его в повседневную жизнь, необходимо:

- триггер — потребность в управлении ПК (например, желание выключить компьютер из другой комнаты);
- действие — установка и настройка приложения;
- результат — удобство, контроль и положительный пользовательский опыт.

Развитие цифровых навыков и потребность в мобильности делают приложения подобного рода особенно актуальными. Люди всё чаще используют мобильные устройства в качестве центра управления — и возможность контролировать ПК с телефона вписывается в этот тренд.

Существует несколько способов реализации подобного функционала:

- через таблицы команд и скриптов (например, вручную заданные сценарии);
- через специализированные приложения, которые обеспечивают более гибкое и визуально удобное управление.

Приложения, в отличие от текстовых сценариев и командных файлов, обеспечивают интуитивно понятный интерфейс, возможность настройки профилей, а также просмотр текущего состояния системы.

Приложение для удалённого управления ПК — это решение, которое позволяет пользователю с мобильного устройства (Android) управлять основными функциями Windows-компьютера. С его помощью можно:

- контролировать медиаплееры;
- выполнять симуляцию действий клавиатуры и мыши;
- запускать и останавливать процессы;
- просматривать текущий статус системы;

– получать уведомления от ПК.

Клиентская часть приложения разрабатывается на языке Java с использованием Android Studio, а серверная часть — для операционной системы Windows в IntelliJ IDEA также на Java. Такое программное решение позволяет реализовать удобную архитектуру клиент-сервер с безопасным и стабильным обменом данными.

Тема разработки подобного ПО является крайне актуальной в современном мире, где важную роль играют мобильность, гибкость, автоматизация и удалённый доступ.

					<i>ПГУ1.090301.16.001 ПЗ</i>	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Анализ аналогов

На данный момент уже представлено много качественных реализаций пультов для ПК, в которых можно выполнять базовые операции с ПК. В данном пункте рассмотрены такие популярные приложения как:

- PC Remote;
- AnyDesk;
- Связь с Windows;
- Unified Remote.

Проведен анализ и выявлены слабые и сильные стороны каждого из приложений.

1.2.1 Мобильное приложение «PC Remote»

Несмотря на серьёзный потенциал и хорошие заделы на будущее, реализация PC Remote пока что недостаточно проработана. Достоинства:

- простота подключения. Приложение быстро и просто нашло ПК в общей локальной сети по Wi-Fi;
- хорошая политика. В базовой версии продукта есть весь необходимый функционал для бытового функционала;
- устройства ввода. Качественная эмуляция работы мыши и клавиатуры, а также возможность рисовать на экране ПК;
- удаленный рабочий стол. Возможность с телефона видеть рабочий стол своего ПК, а также непроработанная возможность добавить виртуальный дисплей (возникла критическая ошибка).

Недостатки:

- некачественная локализация. Не все слова переведены на русский язык;
- некачественный интерфейс. Интуитивно не понятный и запутанный интерфейс;
- плохая реализация функционала. Большинство заявленных функций не работают или перестают работать после первого запуска.

Интерфейс приложения представлен на рисунке 1 и 2.

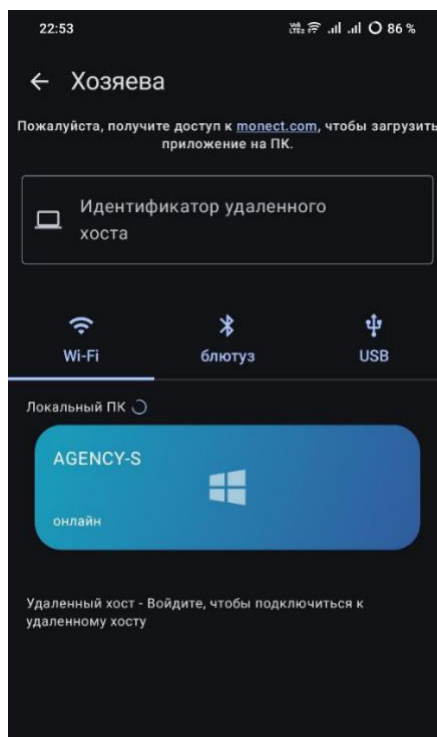


Рисунок 1 - Главная страница приложения

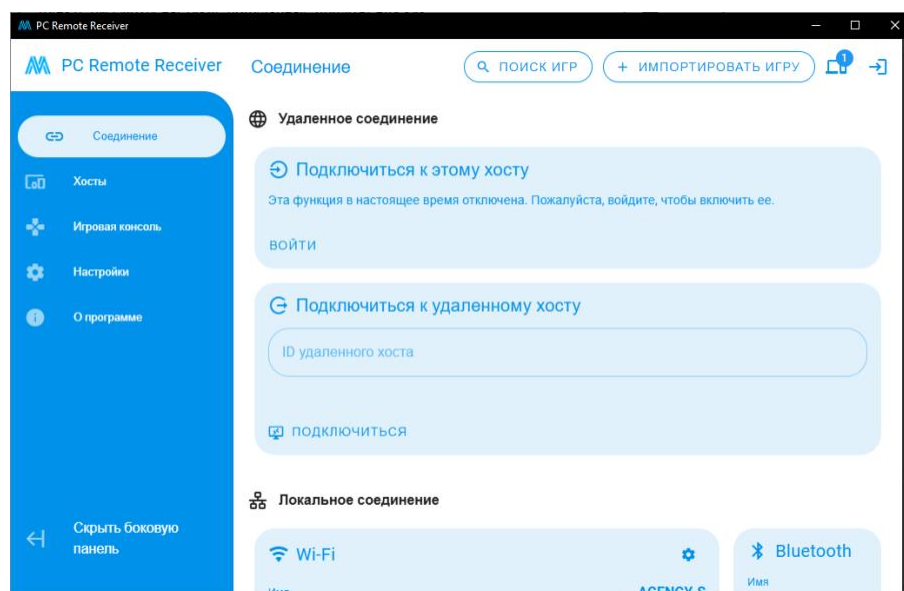


Рисунок 2 – Главная страница настольного приложения

1.2.2 Мобильное приложение «AnyDesk»

Приложение выполнено как офисный пакет для полной удаленной работы с настольными устройствами в формате стриминга.

					ПГУ1.090301.16.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

Достоинства:

- огромный функционал. Данный продукт открывает прямой дистанционный доступ к подключенным устройствам, так как программа эмулирует ввод пользователя напрямую, уничтожая преграду в виде расстояния;
- приятное и понятное обучение. При первом входе в приложение на мобильном устройстве, AnyDesk предлагает пройти обучение, которое при желании можно пропустить. Обучение легкое, красивое и доходчивое;
- возможность подключаться к устройствам по коду. Пользователи могут объединяться и использовать удаленный доступ совместно, используя уникальный код, без необходимости использовать личную учетную запись.

Недостатки:

- тяжеловесная программа. Так как приложение позволяет вести полноценный управляемый стрим, оно сильно нагружает обе системы, что на слабом железе вызывает подтормаживания и глюки, что предоставляет пользователю негативный опыт;
- нет простоты. Так как есть только режим стриминга, нельзя давать быстрые команды «на лету», как, например, в PC Remote, для управления уровнем громкости устройств;
- требовательно к данным. Приложение запрашивает слишком много функционала и конфиденциальных данных, несмотря на дисклеймер при самом первом запуске. Такой повышенный интерес к данным пользователя может вызвать опасения или отвращение к продукту, как было в моем случае.

Интерфейс приложения представлен на рисунке 3 и 4.

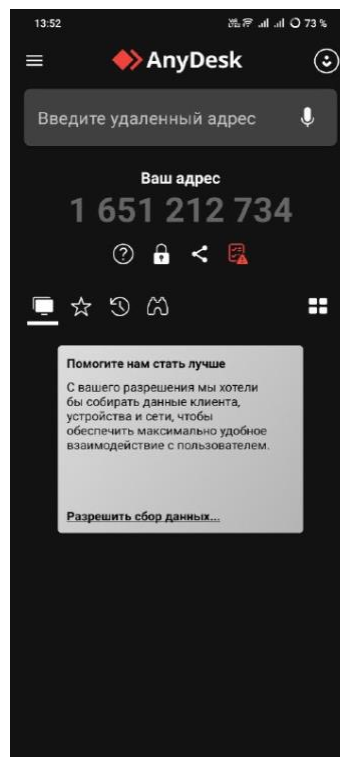


Рисунок 3 – Главная страница с кодом подключения

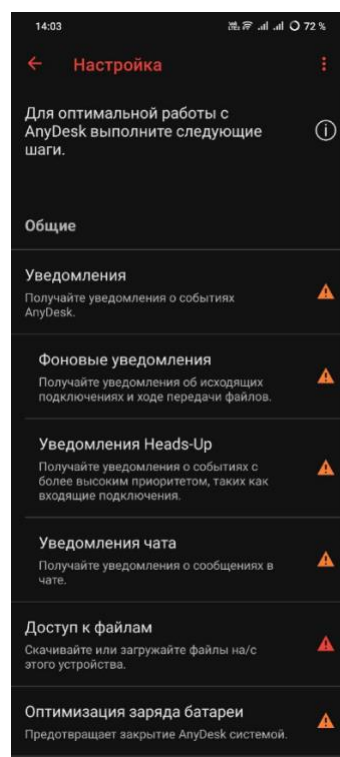


Рисунок 4 - Страница настроек приложения

					ПГУ1.090301.16.001 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

1.2.3 Мобильное приложение «Связь с Windows»

Это приложение было разработано командой Microsoft для обеспечения непрерывной связи устройств на базе ОС Windows с другими устройствами на базе ОС Android и iOS. Приложение позволяет устройствам обмениваться мультимедийными данными, а также принимать на настольных устройствах вызовы и уведомления с мобильных устройств.

Достоинства:

– оригинальный продукт Microsoft, работающий без дополнительной настройки, так как Windows уже поставляется с фоновым процессом «Ваш телефон», что также позволяет приложению работать с минимум тормозов, так как идет прямая интеграция с родными нативными функциями Windows;

– простота использования. Приложение довольно прямолинейно и однозначно направляет пользователя к искомым повседневным задачам;

– довольно приятный дизайн, совпадающий с палитрой Windows, что делает комфортной работу с данным продуктом.

Недостатки:

– узконаправленный функционал; Невозможность обмена иными данными между устройствами (.apk, .exe, .txt и т.д.);

– может не подойти тем, кто не любит контракты, договоры и прочие официальные атрибуты, а также корпорации и компании.

Интерфейс приложения представлен на рисунке 5.

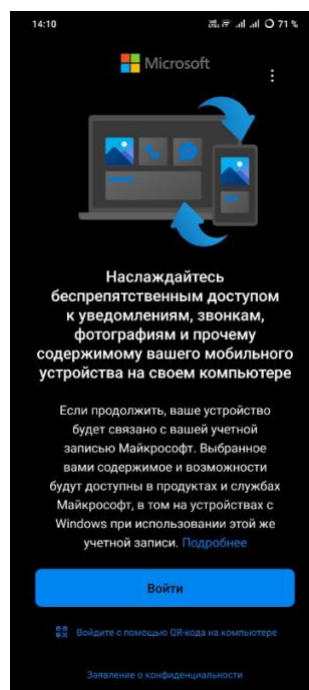


Рисунок 5 - Главная страница входа

1.2.4 Мобильное приложение «Unified Remote»

Данное приложение включает в себя самый полноценный дистанционный пульт управления домашним ПК. В его функционале есть все базовые потребности пользователя, использующего пульт управления ПК.

Достоинства:

- полный необходимый набор функций;
- легкое приложение, понятный интерфейс;
- простота настройки;
- удобство. Приложение простое и не требует дополнительных знаний при использовании.

Недостатки:

- устаревший дизайн приложения. Неприятный зеленый цвет и слишком прямые и большие линии показывают, что приложение давно не обновлялось;
- высокая стоимость полной версии. На 25.04.2025 цена составляет 320 рублей, что представляется завышенной для приложения с ограниченной

					ПГУ1.090301.16.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

функциональностью — например, отсутствием поддержки запуска утилит и скриптов на ПК. Кроме того, приложение не предоставляет прозрачной информации о механизмах передачи данных и уровне обеспечения.

Интерфейс приложения представлен на рисунке 6 и 7.

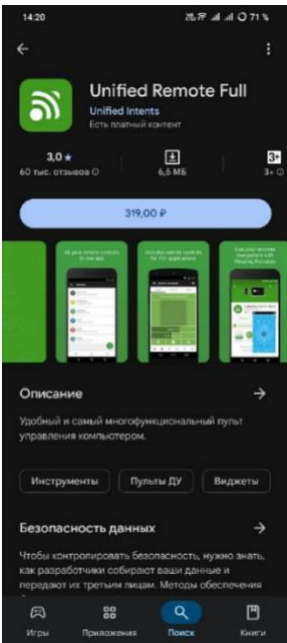


Рисунок 6 – Страница полной версии в магазине

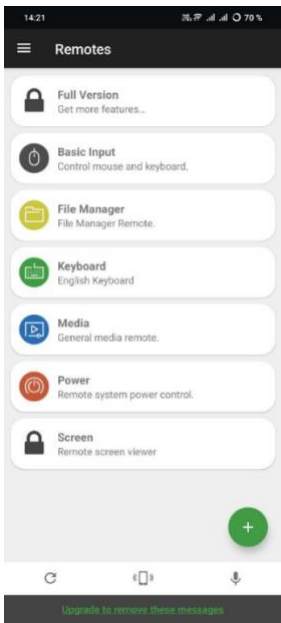


Рисунок 7 – Главное меню приложения

1.2.5 Вывод

Таблица 1 – Достоинства и недостатки приложений-аналогов.

	PC Remote	AnyDesk	Связь с Windows	Unified Remote
Русификация	+	+	+	+
Легкость приложения	+	-	+	+
Удобство	-	-	+	+
Передача любых файлов	+	-	-	+
Безопасность	-	+	+	-
Оригинальная концепция	-	+	+	-
Надежность	-	+	+	-

Анализ существующих решений показывает, что базовый функционал удаленного управления ПК в аналогичных приложениях отличается разнообразием подходов к реализации. Также важно понимать, что приложение — это только удобный способ для достижения цели. Оно должно быть простым, гибким и удобным для самого пользователя, поэтому он сам в праве выбирать более подходящее приложение для себя. Можно заметить, что приложений, специализированных на запуске скриптов и отдачи быстрых и гибких команд на рынке не представлено. Поэтому создание приложения для дистанционного управления ПК по сети является оправданным.

1.3 Постановка задачи

Необходимо разработать приложение для дистанционного управления ПК по сети для настольных и мобильных устройств на базе операционных систем Windows и Android, позволяющие легко и быстро освоить функционал и начать использовать подключенные устройства максимально изящно и эффективно. Согласно статистике, Android и Windows являются неоспоримыми лидерами по количеству смартфонов и настольных компьютеров и занимают 70% от глобального рынка домашних устройств. Именно поэтому для разработки была выбрана эта операционная система.

В приложении должно быть реализовано:

- отображение доступных устройств на главном экране;
- настройку и сохранение IP адресов для подключения к устройствам;
- включение и выключение тост-уведомлений;
- подключение к указанному устройству по TCP протоколу;
- просмотр имени устройства;
- возможность отдачи прямых команд подключенным устройствам (для командной строки CMD Windows);
- возможность изменять положение курсора подключенного устройства и эмулировать нажатие ЛКМ и ПКМ;
- возможность отправлять базовые команды подключенным устройствам (управление уровнем громкости, пауза медиа);
- интуитивно понятный интерфейс с использованием Material 3.

Минимальные системные требования:

- операционная система Android 14;
- наличие локальной сети (для Android наличие модуля WiFi).

Основная цель проекта – предоставить пользователям удобство использования домашних устройств на расстоянии.

2 Выбор инструментов разработки

Для того чтобы разработать качественный продукт и оптимизировать процесс разработки следует выбрать правильные инструменты. Это помогает избежать проблем при производстве и использовании продукта, а также позволяет сократить время и средства на его создание.

2.1 Выбор языка программирования

Android является одной из самых популярных операционных систем для мобильных устройств. Поэтому очень важно выбрать правильный язык программирования, который будет наиболее эффективным инструментом и в полном объеме поможет разработать безопасное и удобное приложение.

Языки, которые позволяют писать приложения для телефонов под управлением Android представлены ниже:

- Java;
- Kotlin;
- C++.

Чтобы решить какой же язык подходит больше всего проведем анализ и выявим особенности каждого из них.

2.1.1 Язык программирования «Java»

Java является официальным языком для разработки программного обеспечения под Android, а также языком который используется в качестве основного в AndroidStudio. Кроме того, Java имеет огромное сообщество разработчиков и большое количество библиотек и фреймворков, что делает этот язык хорошим вариантом для разработки мобильного приложения.

2.1.2 Язык программирования «Kotlin»

Kotlin, как и Java, также является официальным языком для программирования программного обеспечения для мобильных устройств, но с

недавних пор стал предпочтительнее в разработке нежели Java. Однако на более старых версиях Android обертка Kotlin не поддерживается.

2.1.3 Язык программирования «C++»

Из-за своих возможностей и высокой производительности C++ также плотно вошел в мир мобильной разработке, но уже для более узконаправленных приложений таких как 3D-игры или для работы с низкоуровневыми компонентами Android. Но из-за высокой сложности языка, отсутствия сборщика мусора и небезопасных указателей разрабатывать на нем программы становится проблематичной задачей.

2.1.4 Вывод

Проанализировав все варианты, было принято решение выбрать Java в качестве основного языка программирования для разработки приложения для дистанционного управления ПК, потому что, используя его получится написать производительное, широко поддерживаемое и эффективное приложение, а также написать для него серверную часть на Windows без лишних адаптеров.

2.2 Выбор среды разработки

Рассмотрим самую популярную и поддерживаемую IDE для разработки Android приложений – Android Studio, которая была специально разработана и признана официальной средой для программирования программного обеспечения для мобильных устройств. Так как эта IDE была создана на основе IntelliJ IDEA она имеет такой же удобный функционал для редактирования кода, отслеживания файлов в проекте, встроенный эмулятор мобильного устройства и интуитивно понятный интерфейс [4].

Также для разработки мобильных приложений могут использоваться такие IDE как IntelliJ, Eclipse и даже VisualStudio, но именно для разработки приложения на базе операционной системы Android они подходят в меньшей степени, поэтому было решено выбрать Android Studio.

2.3 Выбор базы данных

В качестве импровизированной базы данных были выбраны SharedPreferences и Private Config Files технологии. SharedPreferences - это легковесное хранилище предназначенное для сохранения небольших данных (настройки, флаги и так далее). Данная технология предназначена для хранения строго базовых типов данных: boolean, int, String, float; однако, несмотря на ограниченный набор типов данных, она является максимально удобной и простой в использовании. Private Config Files – система сохранения файлов в приватном хранилище конкретного приложения, что позволяет делать сам Android. Файлы такого типа принадлежат только владельцу и кроме него никто больше не может получить доступ к этим файлам, кроме самого пользователя, однако чаще всего требуются повышенные права доступа root [4].

В SharedPreferences данные хранятся в виде пары «ключ-значение» типа String, что позволяет использовать только одну функцию для асинхронной записи значения – apply(). К сожалению, из-за ограничения типов данных данная технология не подходит для хранения сложных структур данных. Однако, эта технология идеально подходит для хранения краткой информации на подобие флагов, состояний, имен и ключей.

В качестве альтернативы SharedPreferences для решения задачи о хранении большой информации с произвольной структурой было решено использовать Private Config Files, которая позволяет осуществлять полный контроль над структурой файла, является очень гибкой и обеспечивает приватность файлам, делая их недоступными для других приложений. Однако, такая гибкость не позволяет автоматически извлекать необходимые данные без ручного парсинга, работает медленнее SharedPreferences и требует разрешения для внешнего хранилища, начиная с Android 10 со стороны пользователя.

3 Архитектура приложения

3.1 Архитектурные решения в проекте CommanderTask

3.1.1 Принципы организации кода

Проект реализован с применением прагматичного подхода к архитектуре [1], сочетающего:

- логическое разделение ответственности между компонентами;
- инкапсуляцию ключевых функций в специализированных классах;
- гибкость для будущего расширения.

Хотя проект не реализует классическую Clean Architecture, в нём были применены отдельные её принципы там, где это давало оптимальные результаты для разработки и производительности с учетом ограниченного времени разработки.

3.2 Структурные компоненты

3.2.1 Управление конфигурацией (Data Layer)

ConfigManager выступает как централизованный адаптер [3] для работы с настройками:

```
String configData = "ip1=" + ip1 + "\nip2=" + ip2;  
ConfigManager.saveConfig(context, configData);
```

Преимущества:

- изолированный доступ к файловой системе [4];
- единый формат хранения;
- простота тестирования.

3.2.2 Сетевое взаимодействие (Service Layer)

Класс SendCommandTask инкапсулирует [7] логику соединения:

```
try (Socket socket = new Socket(IP, PORT)) {  
    out.writeUTF(command);  
    return in.readLine();  
}
```

Особенности реализации:

- асинхронное выполнение через AsyncTask [2];
- стандартные Java-сокеты для кроссплатформенности;
- обработка ошибок соединения.

3.2.3 Презентационный слой (UI Components)

Активности организованы по принципу специализации:

Таблица 2 - Активности и их ответственности

Активность	Ответственность
ControlActivity	Управление подключенным устройством
DevicesActivity	Выбор и навигация между устройствами
SettingsActivity	Конфигурация параметров приложения

3.3 Ключевые архитектурные решения

3.3.1 Разделение UI и бизнес-логики

Хотя основная логика находится в Activity [4], критические операции вынесены в:

- ConfigManager для работы с данными;
- SendCommandTask для сетевых операций;
- отдельные обработчики для сложной UI-логики.

3.3.2 Паттерн "Адаптер"

Паттерн «Адаптер» [3] реализован в нескольких вариантах:

// 1. Адаптер для хранения настроек
SharedPreferences -> ConfigManager -> Domain Objects

// 2. Адаптер сетевых команд
UI Events -> SendCommandTask -> TCP Protocol

3.3.3 Управление зависимостями

Сознательно использованы подходы, уменьшающие связанность между компонентами:

- явная передача контекста вместо глобальных переменных;
- статические методы только там, где это оправдано;
- локализация Android-зависимостей в UI-слое.

3.4 Возможности для развития архитектуры

Проект специально спроектирован с учетом постепенного перехода к более строгой архитектуре:

3.4.1 Потенциальные улучшения

- Выделение доменного слоя с:
- Device entity-классами;
- DeviceRepository интерфейсом;
- внедрение DI через Dagger/Hilt;
- перевод SendCommandTask в отдельный service-модуль [4].

3.5 Преимущества выбранного подхода

Выбор данной архитектуры имеет следующие преимущества:

- быстрая разработка MVP-версии;
- понятность кода для новичков в команде;
- гибкость изменений за счёт ключевых точек расширения;
- возможность замены ConfigManager на Room/SQLite;
- лёгкий переход с TCP на WebSockets;
- добавление новых Activity без изменения логики.

4 Реализация приложения

Перед началом разработки ПО следует разобраться в архитектуре и функционале разрабатываемого приложения. Так нам потребуется рассмотреть такие важные аспекты разработки приложения как описание работы приложения, функционал, взаимодействие с сервером, а также общую структуру приложения.

4.1 Описание функционала приложения CommanderTask

Основные возможности системы

Приложение CommanderTask предназначено для удалённого управления устройствами через TCP/IP соединение. Для доступа к полному функционалу пользователю необходимо выполнить подключение к целевому устройству. После успешного подключения становятся доступны следующие возможности:

Основные функции управления:

- регулировка уровня громкости (увеличение, уменьшение или полное отключение);
- управление медиаплеером (остановка и пуск воспроизведения);
- контроль курсора мыши (перемещение в четырех направлениях с регулируемой скоростью);
- эмуляция кликов мыши (левый и правый клики);
- отправка произвольных команд;

Функции конфигурации:

- сохранение IP-адресов управляемых устройств [6];
- настройка параметров отображения (включение/отключение toast-уведомлений);
- редактирование сетевых параметров подключения;

Данные варианты использования представлены на диаграмме на рисунке 8.

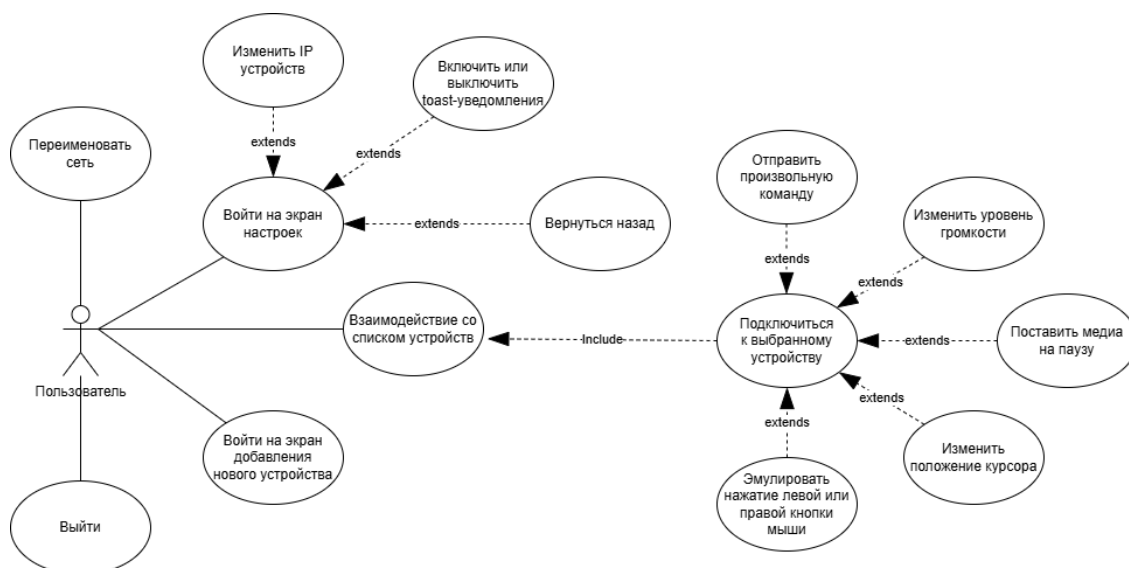


Рисунок 8 – UML-диаграмма вариантов использования для пользователя

Так, пользователь начинает взаимодействие с приложением через его запуск, после чего клиент выводит список сохраненных устройств на главном экране.

Если пользователь нажимает на иконку шестеренки, он переносится на экран настроек, где ему доступно изменение таких параметров, как включение или отключение уведомлений, и адресов подключаемых устройств. После внесения необходимых изменений пользователь может вернуться на главный экран.

На экране доступных устройств пользователю предоставлена возможность нажать на иконку желаемого устройства для подключения к нему и началу отправки команд для дистанционного управления техникой.

Чтобы более точно проанализировать взаимодействие пользователя с приложением составим диаграммы деятельности, представленные на рисунках 9, 10.

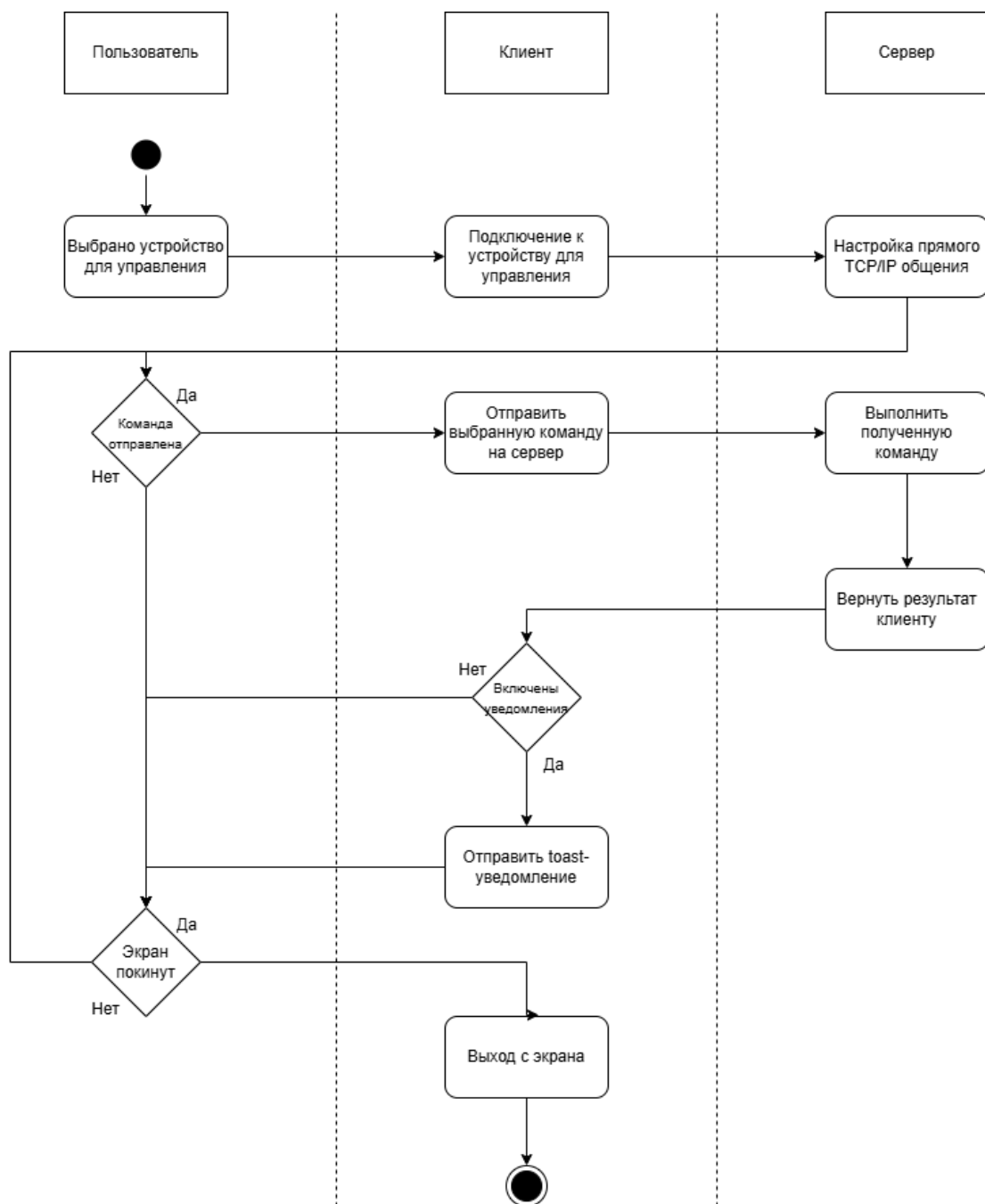


Рисунок 9 – UML диаграмма деятельности для экрана управления

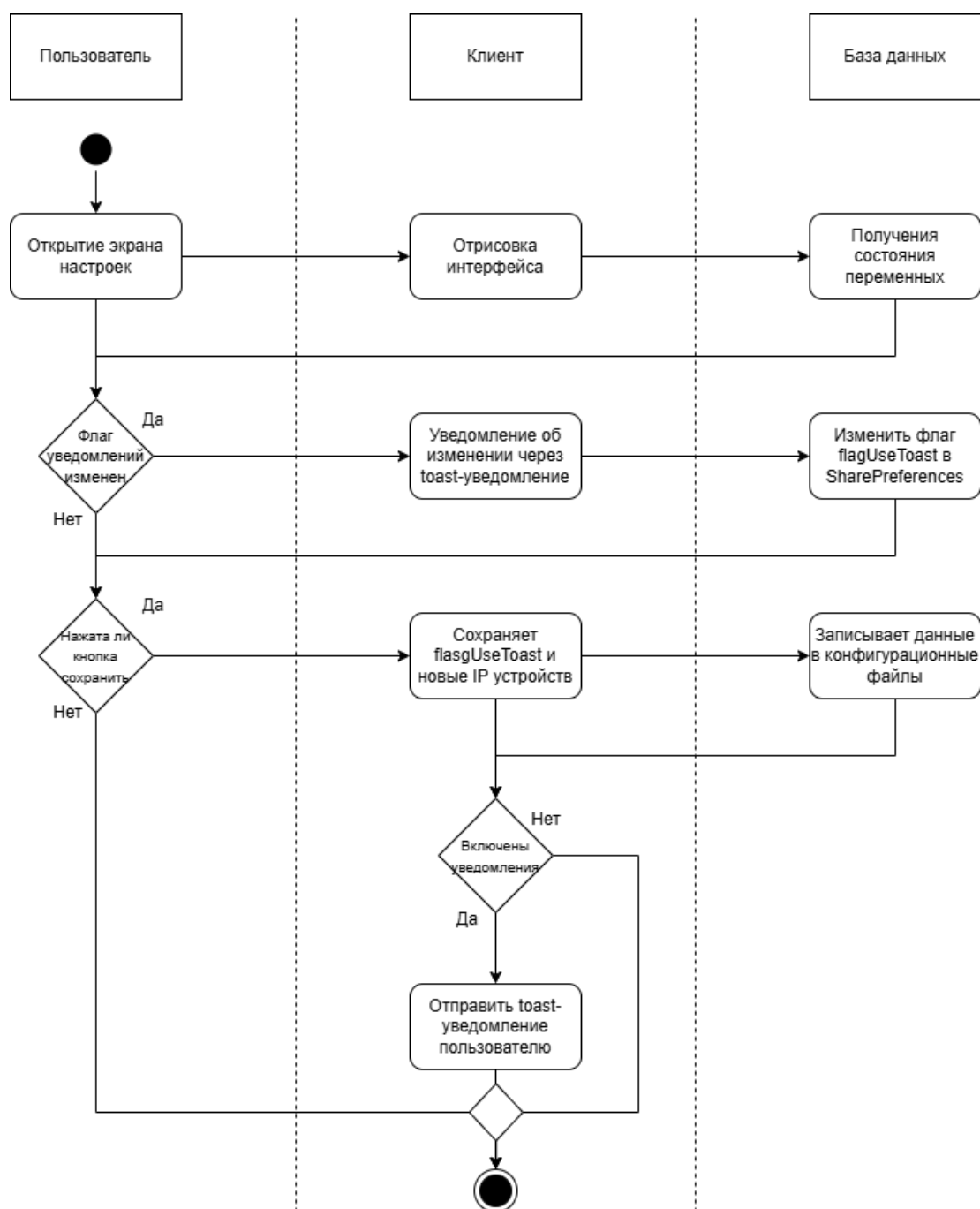


Рисунок 10 – UML диаграмма деятельности для экрана настроек

После перехода на экран управления устройством, клиент запрашивает у сервера его имя и выводит его на экран, в случае неверного или неизвестного имени выводится общее имя "Неизвестное устройство". На данном экране пользователю доступны следующие элементы для взаимодействия: не редактируемое текстовое поле с название устройства, редактируемое текстовое

поле для ввода произвольных команд для более детального взаимодействия с устройством, ползунков установления определенного процента уровня громкости на устройстве, кнопки увеличения и уменьшения уровня громкости, кнопка беззвучного режима, а также пауза медиа на подключенном устройстве. Переходя к интерактивной части пользовательского интерфейса, пользователь может ознакомиться с группой кнопок, направленных на взаимодействие с дистанционным курсором устройства, а именно: кнопки изменения положения (вверх, вниз, влево, вправо) в соответствии со значением ползунка, размещенного над этими кнопками, а также две кнопки, позволяющие эмулировать нажатие важнейших клавиш подключенного устройства: кнопки нажатия левой и правой клавиши мыши.

Данный набор функций является минимально базово необходимым для дистанционного управления подключенным устройством, однако его можно дополнить такими элементами, как: джойстик для более плавного и удобного управления курсором, виртуальная клавиатура для набора текста и эмуляции нажатия особых клавиш, функции для обмена файлами между устройствами в одной сети.

4.2 Описание работы приложения

Пользователь может взаимодействовать как с клиентской частью приложения, так и с серверной [6]. Для того, чтоб начать пользоваться ПО требуется запустить на подходящей версии Android (13+), после чего запустить приложение и подключиться к локальной сети (проводным или беспроводным способом). Далее на экране настроек внести IP адрес для устройств своей сети и сохранить настройки, после чего необходимо запустить сервер на домашних устройствах, которыми пользователь желает управлять, также важно, чтобы эти устройства были подключены к локальной сети, к которой подключен Android-клиент. Далее пользователь выбирает устройство из списка на главном экране

					ПГУ1.090301.16.001 ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

приложения и может отдавать дистанционные команды подключенным устройствам.

Все выполняемые действия как пользователя, так и клиента с сервером изображены на диаграмме последовательности на рисунке 11. А физическое расположение каждого из компонентов данной системы представлено на рисунке 12.

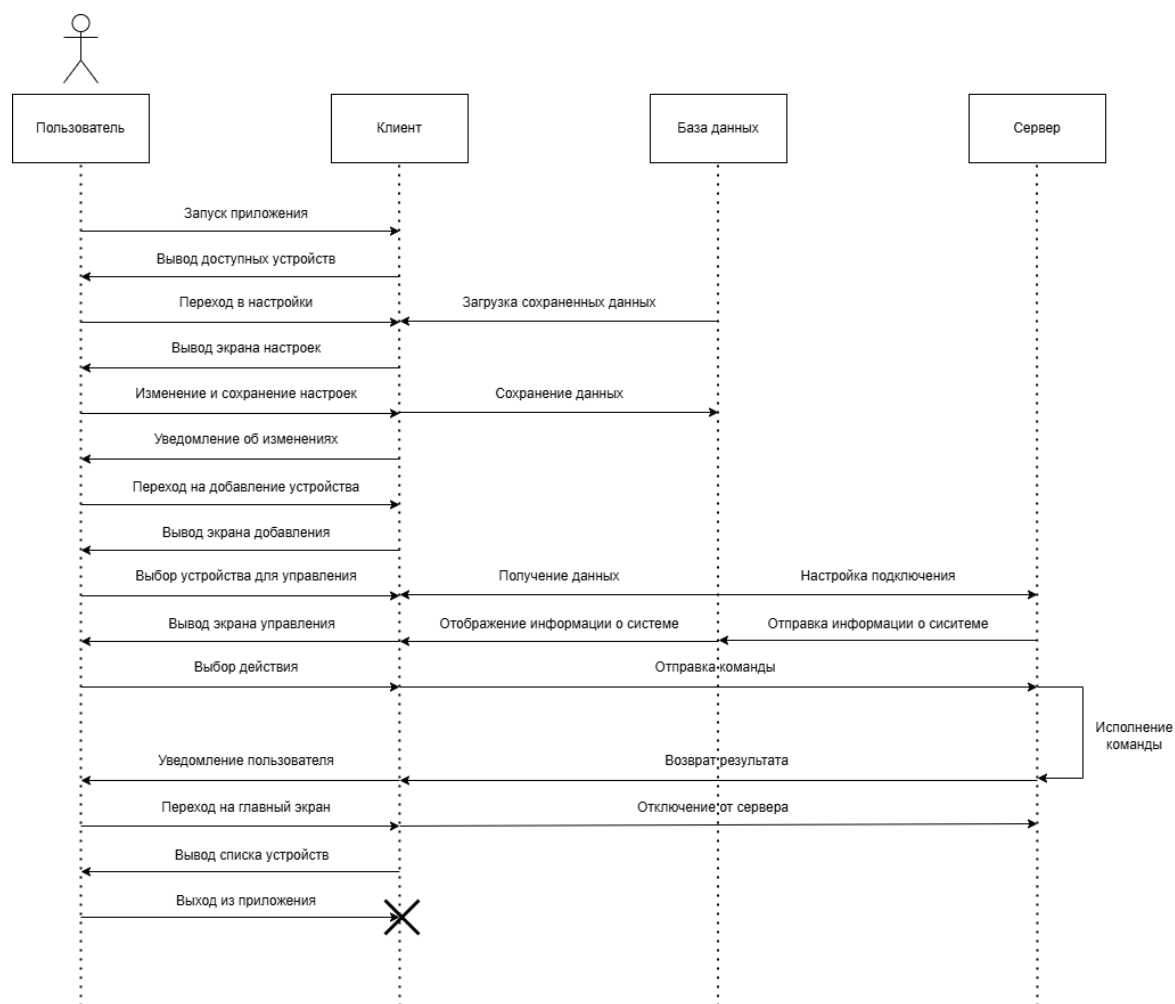


Рисунок 11 – UML-диаграмма последовательности приложения

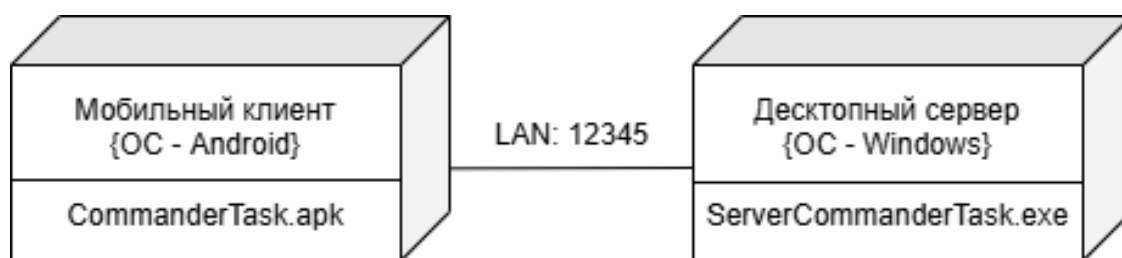


Рисунок 12 – UML-диаграмма развертывания

4.3 Основная реализация приложения

Проект реализован с использованием прагматичного подхода, сочетающего элементы модульности с ориентацией на Android-компоненты. В отличие от классической Clean Architecture [5], структура оптимизирована для быстрой разработки MVP с сохранением ключевых принципов разделения ответственности.

Фактическая структура проекта:

4.3.1 UI-слой (Presentation)

Состав:

- ControlActivity - главный экран управления;
- DevicesActivity - выбор устройства;
- SettingsActivity – настройки;
- AddDeviceActivity - экран добавления.

Особенности:

- логика управления напрямую в Activity;
- отсутствие ViewModel (используются обработчики в UI);
- пример из ControlActivity:

```

sendCommandButton.setOnClickListener(v -> {
    String command = commandInput.getText().toString();
    new SendCommandTask().execute(command); // Прямой вызов
AsyncTask
});
  
```

4.3.2 Бизнес-логика (Domain)

Реализация:

- инкапсулирована в классе `SendCommandTask`;
- правила формирования команд (например, "SET_VOLUME " + `seekBar.getProgress()`);
- обработка сетевых ошибок:

```
String command = "MOVE_CUR_LEFT " + speedCursor;  
new SendCommandTask().execute(command);
```

4.3.3 Слой данных (Data)

Компоненты:

- `ConfigManager` - работа с конфигурацией;
- `SharedPreferences` - хранение настроек.

Пример сохранения:

```
String configData = "ip1=" + ip1 + "\nip2=" + ip2;  
ConfigManager.saveConfig(context, configData);
```

4.4 Ключевые отличия от Clean Architecture

4.4.1 Смешение слоёв

`ControlActivity` сочетает UI, логику и сетевые вызовы, отсутствует четкое разделение на `UseCases`.

4.4.2 Управление зависимостями

В программе присутствует прямое создание экземпляров (`new SendCommandTask()`), используется вызов статических таких методов, как (`ConfigManager.saveConfig()`).

4.4.3 Работа с данными

Пример потока данных по сценарию: Изменение громкости

- пользователь двигает `SeekBar` в `ControlActivity`;
- `SendCommandTask` отправляет команду через сокет;
- сервер выполняет действие и возвращает ответ.

4.5 Преимущества выбранного подхода

Быстрая разработка:

- минимальное количество классов;
- прямой доступ к Android API.

Понятность кода:

- логика сосредоточена в Activity;
- явные вызовы без сложных абстракций.

Гибкость изменений:

- легкая модификация команд;
- простое добавление новых UI-элементов.

Чтобы совсем не запутаться в классах и их методах посмотрим на рисунок 13, на котором представлена диаграмма классов.

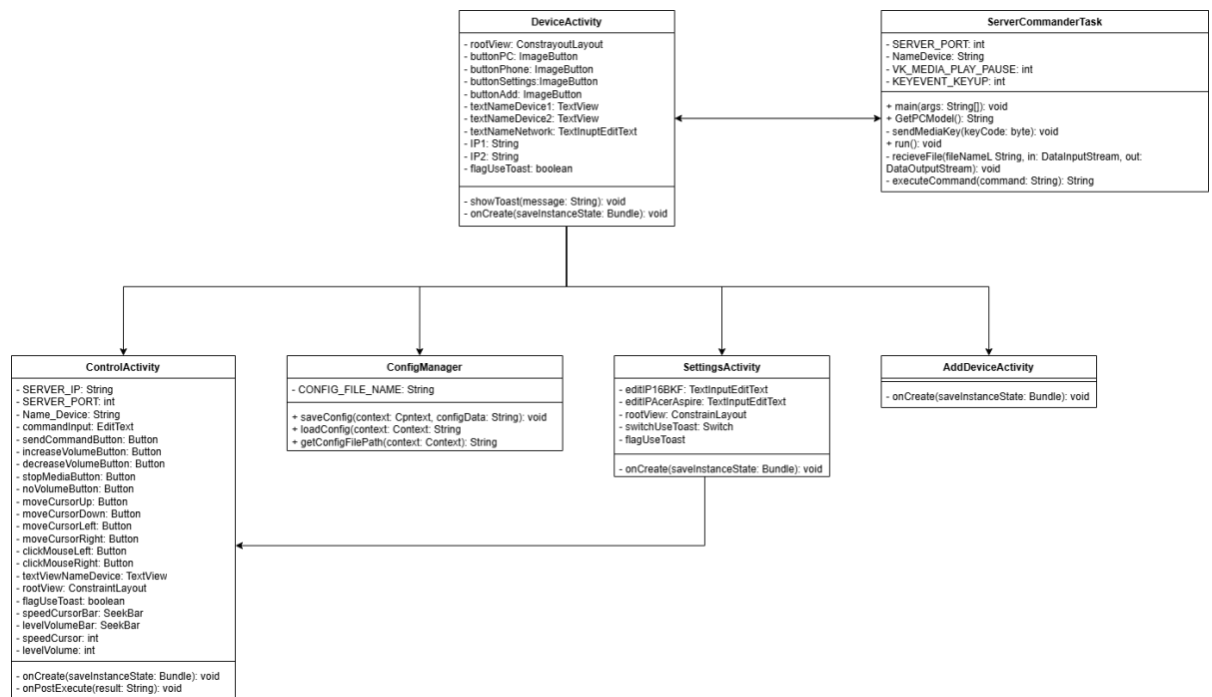


Рисунок 13 - Диаграмма классов клиента и сервера

5 Тестирование

После того как получилось реализовать первый прототип приложения, следует заняться его тестированием. Этот этап является одним из важнейших этапов при разработке ПО, так как помогает выявить проблемы и ошибки в коде, а также улучшить пользовательский опыт взаимодействия.

В данном пункте будут протестированы такие программные моменты как:

- настройка приложения;
- просмотр списка устройств;
- подключение к устройству;
- отправка команды устройству;
- выполнение команды на сервере.

5.1 Взаимодействие с главным экраном

При открытии приложения, пользователю будет показан список доступных для подключения устройств в локальной сети (рисунок 14). Для работы приложения необходимо так же запустить серверное приложение на управляемом устройстве (рисунок 15).

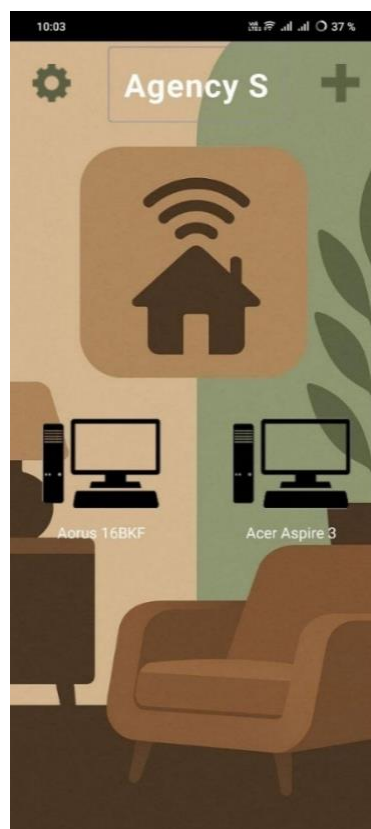


Рисунок 14 – Главный экран приложения

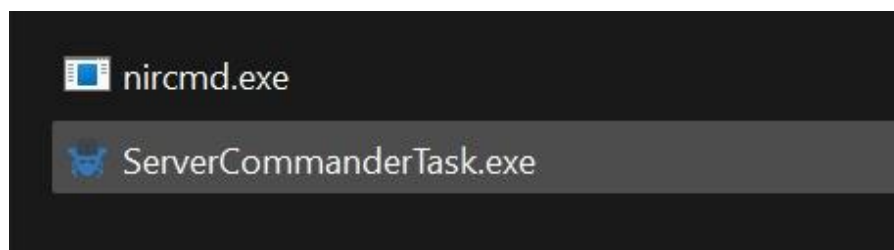


Рисунок 15 – Серверное приложение

На данном экране присутствует редактируемое поле – название сети, в которой находятся устройства, две служебные кнопки – шестеренка для перехода на экран настроек и плюсик – для перехода на экран добавления нового устройства. Так же по центру представлены две плитки устройств, нажав на которые, мы можем перейти к управлению ими.

Пользователь может изменить название текущей сети (рисунок 16), перейти на экран настроек (рисунок 17), экран добавления нового устройства (рисунок 18).

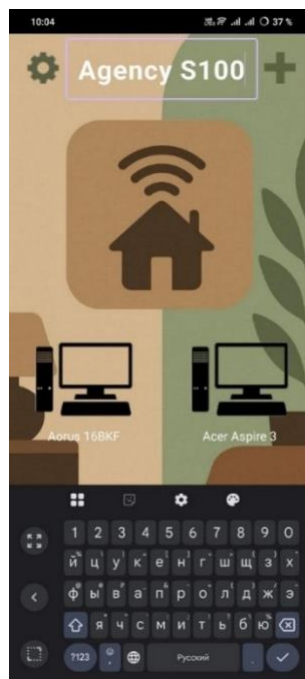


Рисунок 16 – Изменение названия сети

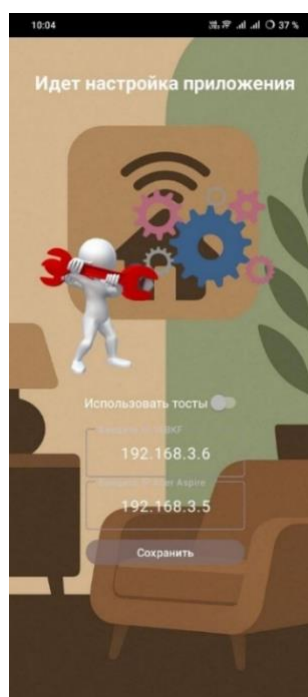


Рисунок 17 – Экран настроек

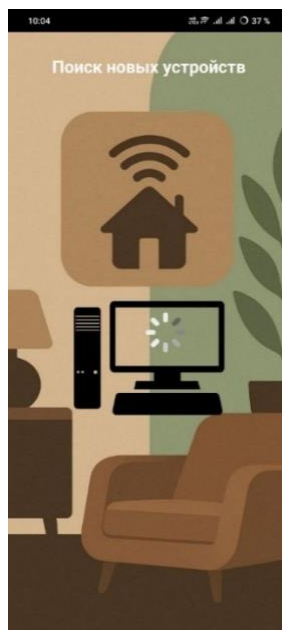


Рисунок 18 – Экран добавления устройства

После того как пользователь нажмет на плитку подключенного устройства, он перейдет на экран управления подключенным устройством (рисунок 19).



Рисунок 19 – Выбрано устройство для управления

5.2 Отправка команд

Если выбрать ввод произвольной команды, например запуска определенного приложения, то сервер пришлет ответ о выполнении команды (рисунок 20).



Рисунок 20 – Уведомление о выполнении команды

Чтобы проверить выполнение команды перейдем на сервер, где обнаружим открытое приложение Блокнот, которое запустит сервер по команде нашего клиента (рисунок 21).

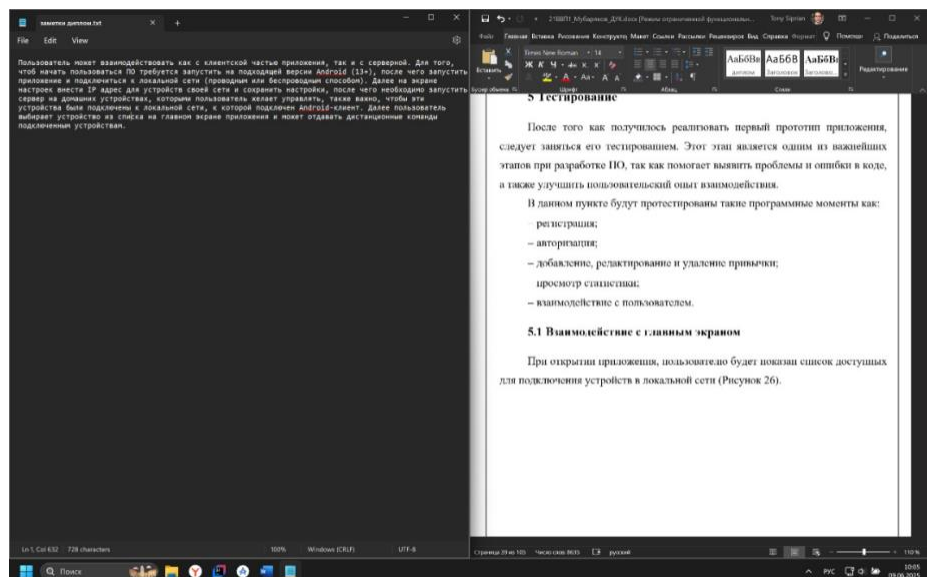


Рисунок 21 – Результат выполнения команды

Подвинув ползунок для изменения уровня громкости в приложении, пользователь на подключенном устройстве получит быстрый результат – уровень громкости на ведомом ПК изменился на заданный (рисунки 22, 23).



Рисунок 22 – Отправлена команда задать уровень громкости 70

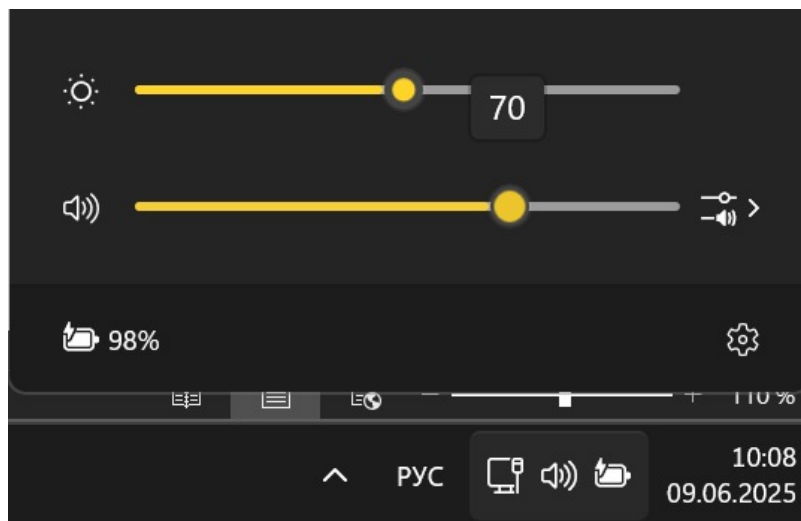


Рисунок 23 – Команда выполнена

Наконец, попробуем сменить положение курсора на экране и нажать левую кнопку мыши (рисунки 24 - 26).



Рисунок 24 – Выполнение команды сместить курсор на заданное расстояние



Рисунок 25 – Отправлена команда нажать левую кнопку мыши

Проверим выполнение команд на сервере (рисунок 26).

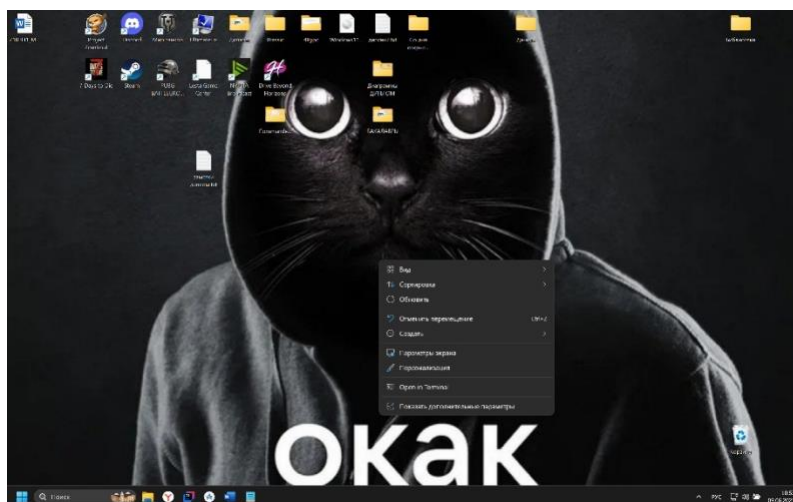


Рисунок 26 – Выполнение на сервере

					ПГУ1.090301.16.001 ПЗ	Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		

5.3 Изменение настроек

Если в настройках убрать показ toast-уведомлений (рисунок 27), то всплывающие уведомления при каждом действии перестанут появляться (рисунок 28).

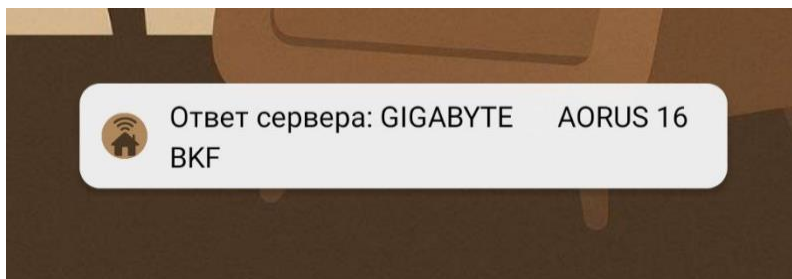


Рисунок 27 – Отключение уведомлений в приложении

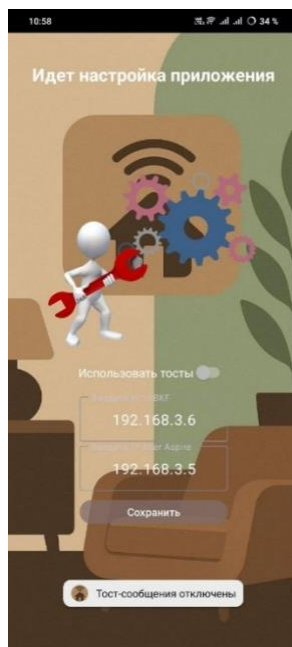


Рисунок 28 – Отключение уведомлений в приложении

Если пользователь сохранит изменения, то появится последнее уведомление, что информация сохранена и больше уведомления появляться не будут (рисунки 29, 30).

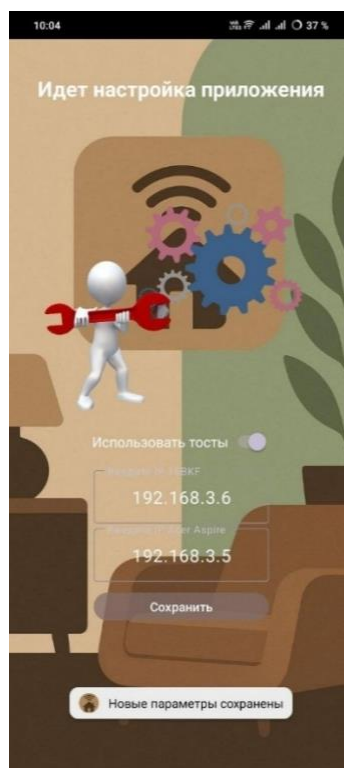


Рисунок 29 – Результат сохранения настроек



Рисунок 30 – Отсутствие уведомлений

5.4 Вывод

В момент создания приложения было выявлено достаточное количество ошибок. Но протестировав ПО получилось исправить и обработать большую часть из них, это позволило сделать более надежное и отказоустойчивое приложение.

					<i>ПГУ1.090301.16.001 ПЗ</i>	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

6 Безопасность жизнедеятельности

Чтобы максимально обезопасить жизни разработчиков и обычных людей, которые проводят за компьютером достаточно много времени, были разработаны специальные правила и нормативы по обеспечению безопасности за ПК и мобильными устройствами. Пренебрежение данными правилами может влечь за собой негативные последствия как для здоровья пользователя, так и для его имущества.

В данном пункте [8] будут рассмотрены основные правила и инструкции по работе человека с компьютером и мобильным телефоном.

6.1 Правила и техника безопасности при работе с ПК

Человек при взаимодействии с вычислительной техникой может подвергаться таким вредным и опасным воздействиям как:

- поражение электрическим током;
- электромагнитное излучение;
- опасность пожара;
- повышенный уровень шума;
- механические повреждения.

Для того чтобы снизить данное влияние требуется соблюдать ряд определенных правил.

В первую очередь нужно обеспечить рабочее помещение достаточным уровнем освещения, которое не будет оставлять бликов на мониторе и светить прямо в глаза рабочему. Нужно учитывать, что освещение должно быть, как естественным, так и искусственным, но не слишком ярким.

Также требуется обзавестись удобной и эргономичной офисной мебелью. Стул должен регулироваться по высоте и иметь регулировку спинки, чтобы ноги могли быть выпрямлены вперед и ступни полностью стояли на полу, а также для того, чтобы работник мог опереться в спинку стула. Разработчик должен сидеть

					ПГУ1.090301.16.001 ПЗ	Лист
						46
Изм.	Лист	№ докум.	Подпись	Дата		

от монитора на уровне вытянутой руки, а монитор должен быть наклонен примерно на 30 градусов. Все эти правила по безопасности позволят обеспечить правильную осанку, а также снизить нагрузку на позвоночник, руки и голову.

В целях безопасности не следует пользоваться компьютером или любой техникой при наличии внешних повреждений. Необходимо следить на своем рабочем месте за исправным состоянием техники и не оставлять ее включенной на долго без присмотра людей. Рабочие места должны быть установлены вдоль стен так, чтобы монитор задней стенкой был обращен к стене. Это поможет снизить количество вредных электромагнитных излучений.

Чтобы снизить как физическую, так и психологическую нагрузку требуется периодически отдыхать от работы. При 8 часовой смене общая продолжительность перерывов должна составлять от 30 до 90 минут. Также для профилактики заболевания глаз и опорно-двигательного аппарата работнику следует проводить короткие упражнения, нацеленные на снятия нагрузок с собственного тела.

6.2 Правила и техника безопасности при работе со смартфоном

Так как основным устройством, где будет работать приложение, является телефон, то следует также проанализировать и учесть все правила безопасности по обращению с ним.

Не следует пытаться вскрывать, разбирать или модифицировать устройство самостоятельно. Также не следует подвергать устройство физическим нагрузкам (ронять, сгибать, бросать, протыкать) или воздействию огня или воды.

Если мобильное или зарядное устройство неисправно использовать его строго запрещено. Также следует пользоваться только оригинальными зарядными устройствами и аксессуарами.

Мобильное устройство постоянно выделяет радиоизлучение, поэтому в целях безопасности не стоит класть телефон близко к жизненно важным органам, а также при разговоре периодически менять положение смартфона относительно головы. Также нужно учитывать, что данных и качественных исследований относительно пагубного влияния радиоизлучения пока что нет.

Последнее на что следует обратить внимание это на чрезмерное злоупотребление использованием телефона может привести к зависимости, а также нарушить сон и концентрацию.

					<p><i>ПГУ1.090301.16.001 ПЗ</i></p>	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

Результатом данной выпускной квалификационной работы стало успешно разработанное мобильное приложение CommanderTask, полностью соответствующее техническому заданию. В ходе работы были выполнены следующие ключевые задачи:

- проведен анализ существующих решений для удалённого управления (TeamViewer, AnyDesk, RDP) и выявлены их ограничения;
- обоснован выбор технологий: Java/Android SDK для клиента, TCP-сокеты для связи;
- реализована модульная архитектура с разделением на:
 - UI-слой (Activities);
 - сетевой модуль (SendCommandTask);
 - конфигурационный менеджер (ConfigManager);
 - разработан протокол обмена текстовыми командами (SET_VOLUME, MOVE_CUR и др.).

Проведено комплексное тестирование:

- юнит-тесты ConfigManager;
- интеграционные тесты TCP-соединения;
- UI-тесты на различных устройствах.

Приложение демонстрирует:

- стабильную работу при сетевой задержке < 100 мс;
- минимальное потребление ресурсов;
- интуитивно понятный интерфейс;

Проект готов к практическому применению для:

- удалённого администрирования ПК;
- управления медиацентрами;
- образовательных целей (изучение сетевого программирования).

Список использованных источников

1. Блох Дж. Effective Java [Текст]. – 3-е изд. – СПб.: Питер, 2019. – 412 с.
2. Хорстманн К. Java. Библиотека профессионала. Том 1. Основы [Текст]. – М.: Вильямс, 2022. – 864 с.
3. Мартин Р. Чистый код. Создание, анализ и рефакторинг [Текст]. – СПб.: Питер, 2020. – 464 с.
4. Официальная документация Android (Java) [Электронный ресурс]. – URL: <https://developer.android.com/docs> (дата обращения: 15.06.2024).
5. Java SE 17 Documentation [Электронный ресурс]. – URL: <https://docs.oracle.com/en/java/javase/17/> (дата обращения: 15.06.2024).
6. TCP/IP Socket Programming in Java [Электронный ресурс]. – URL: <https://www.baeldung.com/a-guide-to-java-sockets> (дата обращения: 15.06.2024).
7. Android Networking Best Practices [Электронный ресурс]. – URL: <https://developer.android.com/training/basics/network-ops> (дата обращения: 15.06.2024).
8. ГОСТ Р 7.0.100-2018. Библиографическая запись. Библиографическое описание [Текст]. – М.: Стандартинформ, 2018. – 52 с.

Приложение А

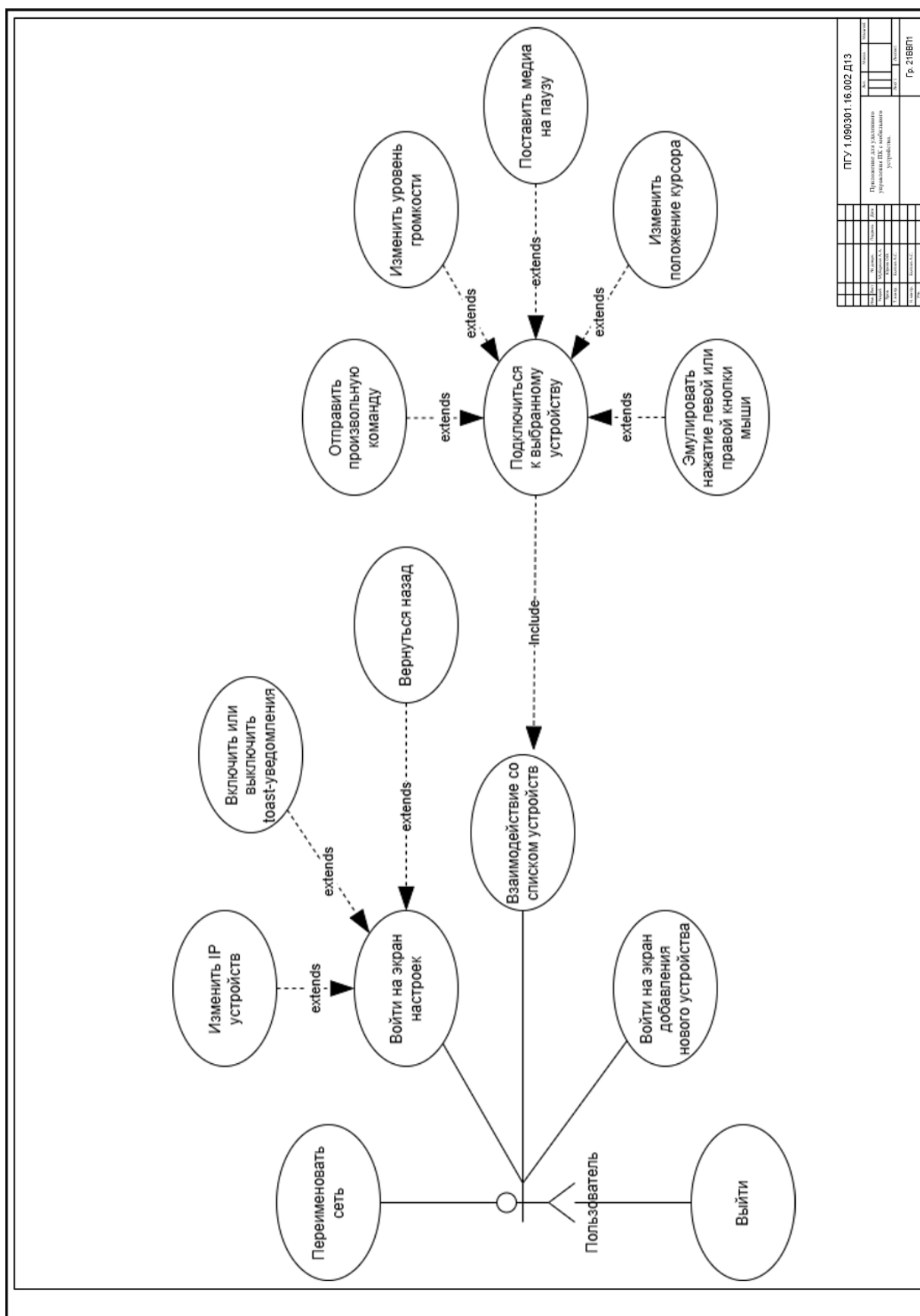
(обязательное)

Приложение для удаленного управления ПК с мобильного устройства

UML–диаграммы

					<i>ПГУ1.090301.16.001 ПЗ</i>	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

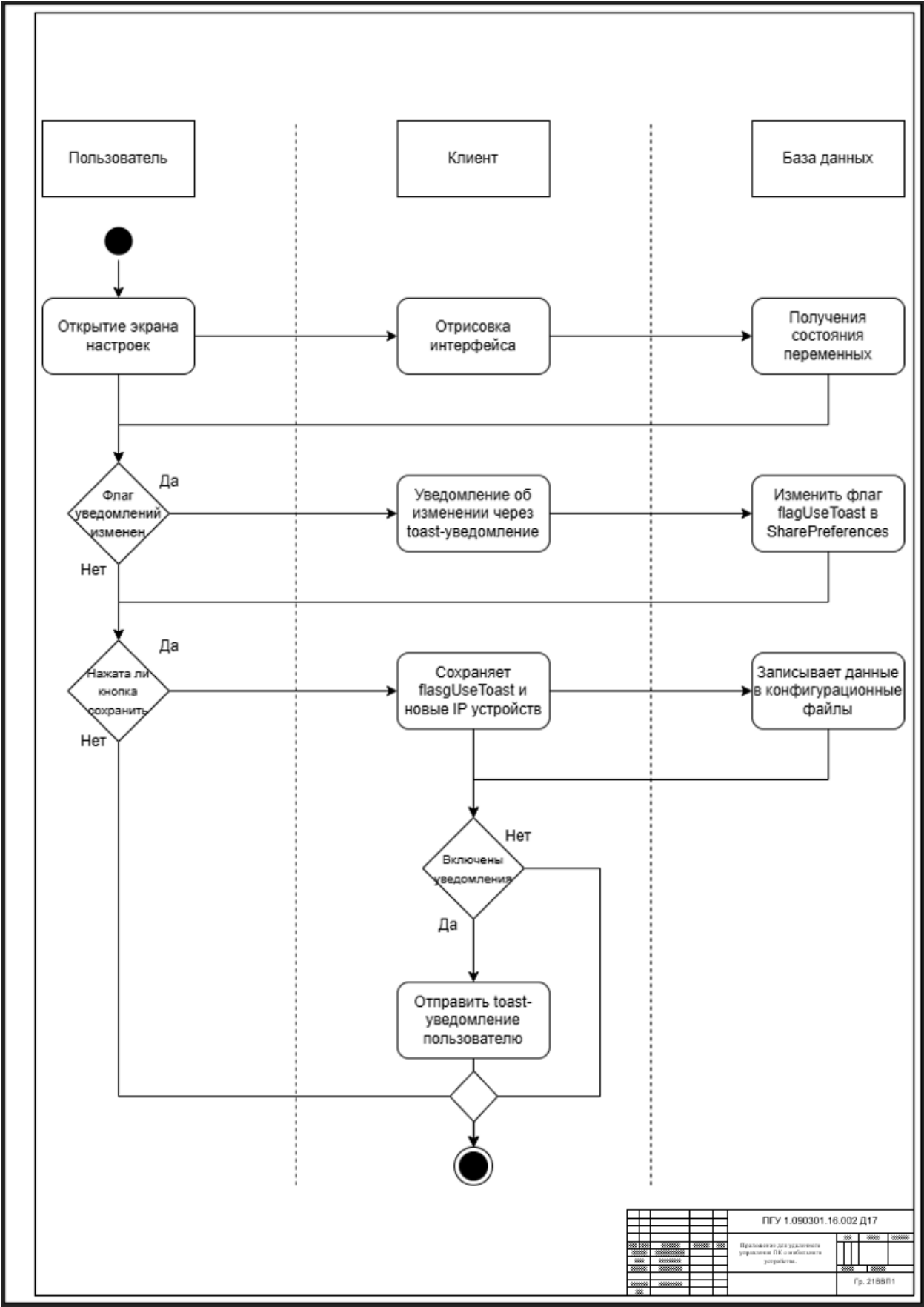
Приложение А.1-Диаграмма прецедентов

[illegible]

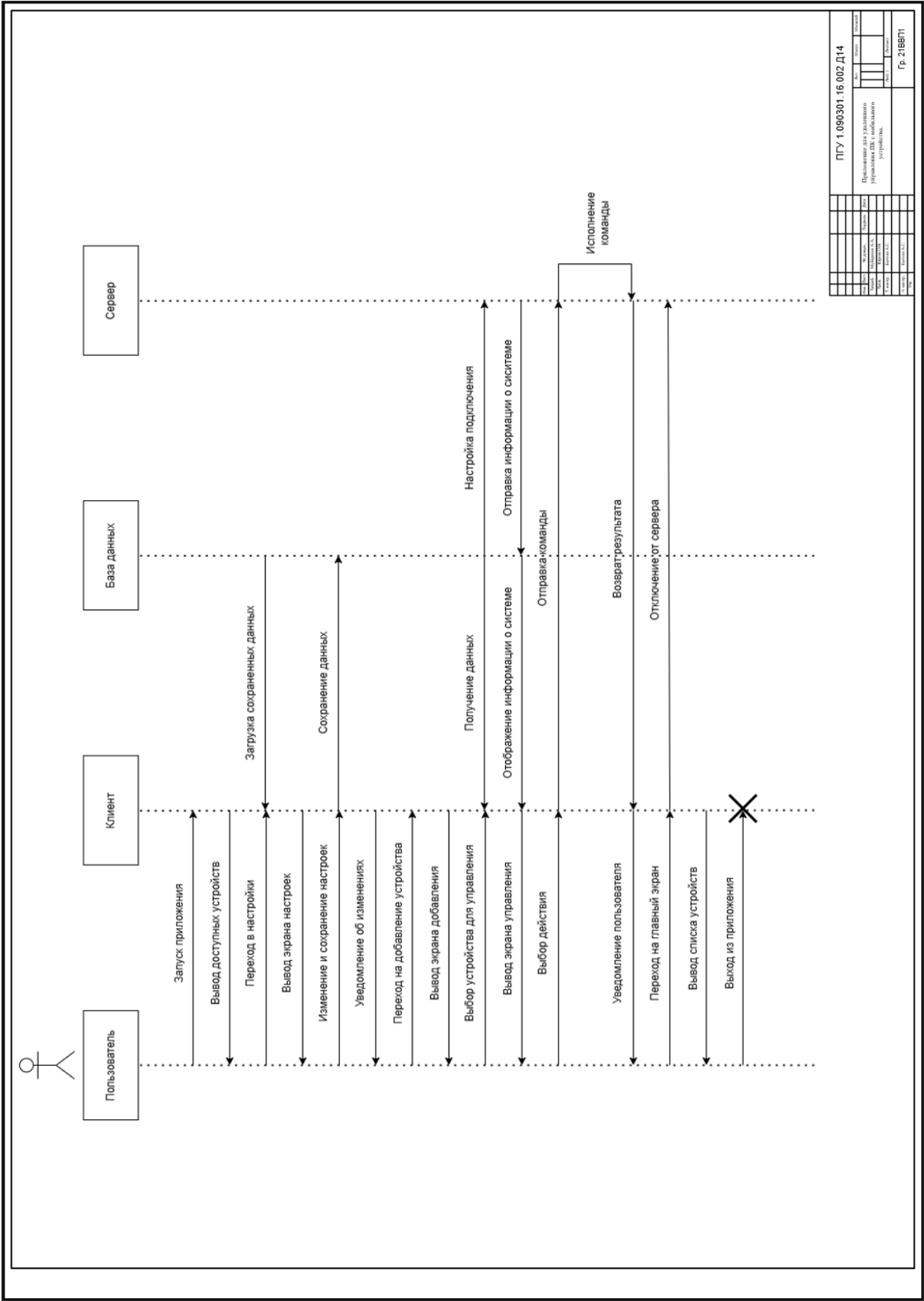
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.16.001 ПЗ

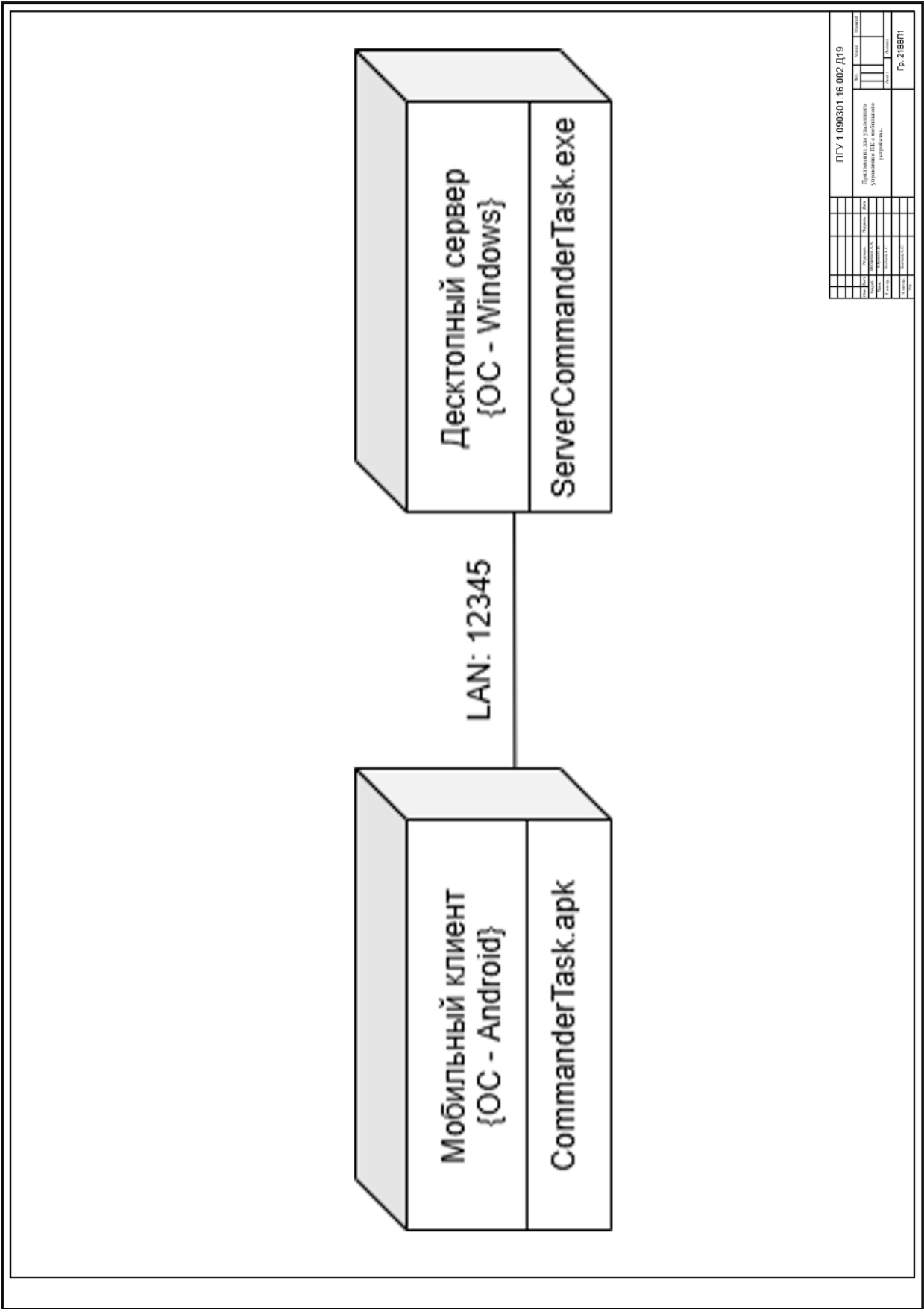
Приложение А.2-Диаграмма деятельности экрана настроек



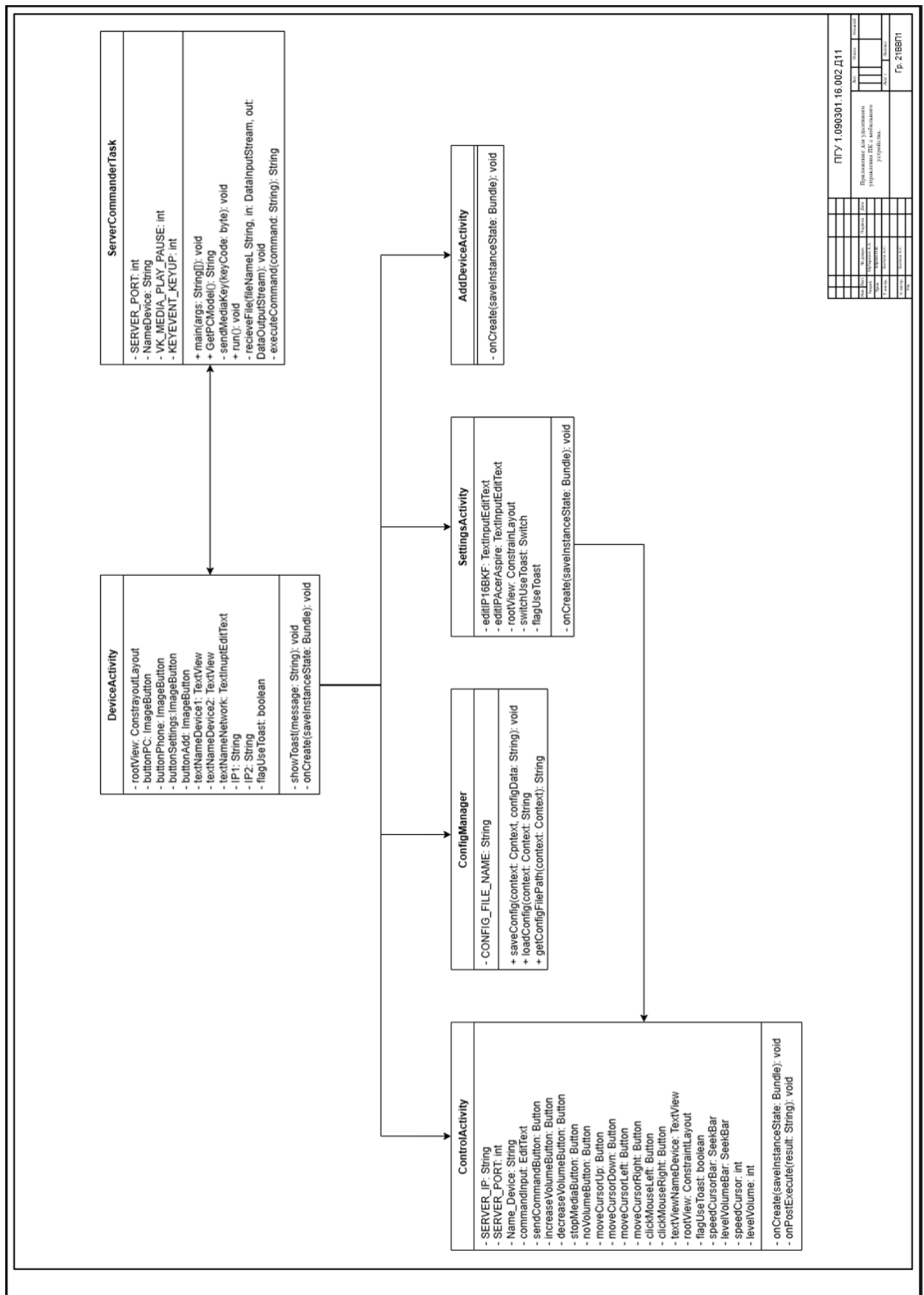
Приложение А.3-Диаграмма последовательности



Приложение А.4-Диаграмма развертывания



Приложение А.5-Диаграмма классов



Приложение Б

(обязательное)

Приложение для удаленного управления ПК с мобильного устройства

Листинг программы

					<i>ПГУ1.090301.16.001 ПЗ</i>	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

Файл «DevicesActivity.java»

```
package com.example.commandertask;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import com.google.android.material.textfield.TextInputEditText;

public class DevicesActivity extends AppCompatActivity {
    ConstraintLayout rootView;

    ImageButton buttonPC;
    ImageButton buttonPhone;
    ImageButton buttonSettings;
    ImageButton buttonAdd;

    TextView textNameDevice1;
    TextView textNameDevice2;

    TextInputEditText textNameNetwork;

    String IP1, IP2;
    boolean flagUseToast;

    private void showToast(String message) {
        // Чтение (в любом Activity)
        SharedPreferences prefs =
getSharedPreferences("AppPrefs", MODE_PRIVATE);
        flagUseToast = prefs.getBoolean("use_toast", false);
        if(!flagUseToast) return;
        Toast.makeText(this, message,
Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_devices);
    }
}
```

					ПГУ1.090301.16.001 ПЗ	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		

```

        // Загрузка конфига с проверкой на null
        String config =
        ConfigManager.loadConfig(DevicesActivity.this);
        if (config != null && !config.isEmpty()) {
            try {
                String[] lines = config.split("\n");
                if (lines.length >= 2) {
                    IP1 = lines[0].replace("ip1=", "").trim();
                    IP2 = lines[1].replace("ip2=", "").trim();
                }
                Log.d("ConfigLoad", "Data load: " + config);
            } catch (Exception e) {
                Log.e("ConfigLoad", "Error parsing config", e);
            }
        }

        rootView = findViewById(R.id.constraint);

        buttonPC = findViewById(R.id.imageButton);
        buttonPhone = findViewById(R.id.imageButton2);
        buttonSettings = findViewById(R.id.imageButton3);
        buttonAdd = findViewById(R.id.imageButton4);

        textNameDevice1 = findViewById(R.id.textView2);
        textNameDevice2 = findViewById(R.id.textView3);

        textNameDevice1.setText("Aorus 16BKF");
        textNameDevice2.setText("Acer Aspire 3");

        textNameNetwork = findViewById(R.id.textNameNetwork);

        // Вешаем обработчик клика
        rootView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Убираем фокус с EditText
                textNameNetwork.clearFocus();

                // Скрываем клавиатуру
                InputMethodManager imm = (InputMethodManager)
                getSystemService(Context.INPUT_METHOD_SERVICE);
                if (imm != null) {

                    imm.hideSoftInputFromWindow(textNameNetwork.getWindowToken(),
                    0);
                }
            }
        });
    });

```

```

// Добавить OnClickListener
buttonPC.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Действие при клике
        showToast("Выбран Aorus 16BKF для управления");

        // Создаем Intent для перехода на
ControlActivity
        Intent intent = new Intent(DevicesActivity.this,
ControlActivity.class);

        // Передаем данные через Intent (можно
передавать строки, числа, объекты и т. д.)
        intent.putExtra("IP_Address", IP1);
        intent.putExtra("Port", 12345);
        intent.putExtra("Name_Device", "Aorus 16BKF");

        // Запускаем ControlActivity
        startActivity(intent);
    }
});

// Добавить OnClickListener
buttonPhone.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Действие при клике
        showToast("Выбран Acer Aspire 3 для
управления");

        // Создаем Intent для перехода на
ControlActivity
        Intent intent = new Intent(DevicesActivity.this,
ControlActivity.class);

        // Передаем данные через Intent (можно
передавать строки, числа, объекты и т. д.)
        intent.putExtra("IP_Address", IP2);
        intent.putExtra("Port", 12345);
        intent.putExtra("Name_Device", "Acer Aspire 3");

        // Запускаем ControlActivity
        startActivity(intent);
    }
});

// Добавить OnClickListener
buttonSettings.setOnClickListener(new

```

```

View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Действие при клике
        showToast("Экран настроек");

        // Создаем Intent для перехода на
SettingsActivity
        Intent intent = new Intent(DevicesActivity.this,
SettingsActivity.class);

        // Запускаем SettingsDeviceActivity
        startActivity(intent);
    }
});

// Добавить OnClickListener
buttonAdd.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        // Действие при клике
        showToast("Экран добавления устройств");

        // Создаем Intent для перехода на
AdddeviceActivity
        Intent intent = new Intent(DevicesActivity.this,
AddDeviceActivity.class);

        // Запускаем AdddeviceActivity
        startActivity(intent);
    }
});
}
}

```

Файл «SettingsActivity.java»

```

package com.example.commandertask;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;

```

					ПГУ1.090301.16.001 ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

```

import android.widget.ImageView;
import android.widget.Switch;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import com.bumptech.glide.Glide;
import com.google.android.material.textfield.TextInputEditText;

public class SettingsActivity extends AppCompatActivity {

    TextInputEditText editIP16BKF;
    TextInputEditText editIPacerAspire;
    ConstraintLayout rootView;
    Button buttonSaveIP;
    Switch switchUseToast;

    boolean flagUseToast;

    private void showToast(String message) {
        // Чтение (в любом Activity)
        SharedPreferences prefs =
getSharedPreferences("AppPrefs", MODE_PRIVATE);
        flagUseToast = prefs.getBoolean("use_toast", false);
        if(!flagUseToast) return;
        Toast.makeText(this, message,
Toast.LENGTH_SHORT).show();
    }

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);

        rootView = findViewById(R.id.con_settings);
        editIP16BKF = findViewById(R.id.ip16bkf);
        editIPacerAspire = findViewById(R.id.ipacer);
        buttonSaveIP = findViewById(R.id.SaveIP);
        switchUseToast = findViewById(R.id.switchUseToast);

        // Чтение (в любом Activity)
        SharedPreferences prefs =
getSharedPreferences("AppPrefs", MODE_PRIVATE);
        switchUseToast.setChecked(prefs.getBoolean("use_toast",
false));
        flagUseToast = switchUseToast.isChecked();

        ImageView gifImageView =
findViewById(R.id.gifSettings1);

        Glide.with(this).load(R.drawable.settings1).into(gifImageView);

```

```

        ImageView gifImageView2 =
findViewById(R.id.gifSettings2);

Glide.with(this).load(R.drawable.settings2).into(gifImageView2);

        // Загрузка конфига с проверкой на null
        String config =
ConfigManager.loadConfig(SettingsActivity.this);
        if (config != null && !config.isEmpty()) {
            try {
                String[] lines = config.split("\n");
                if (lines.length >= 2) {
                    String ip1 = lines[0].replace("ip1=",
"").trim();

                    String ip2 = lines[1].replace("ip2=",
"").trim();

                    editIP16BKF.setText(ip1);
                    editIPacerAspire.setText(ip2);
                }
            } catch (Exception e) {
                Log.e("ConfigLoad", "Error parsing config", e);
            }
        }

        // Вешаем обработчик клика
        rootView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Убираем фокус с EditText
                editIP16BKF.clearFocus();
                editIPacerAspire.clearFocus();

                // Скрываем клавиатуру
                InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
                if (imm != null) {

                    imm.hideSoftInputFromWindow(editIP16BKF.getWindowToken(), 0);

                    imm.hideSoftInputFromWindow(editIPacerAspire.getWindowToken(),
0);
                }
            }
        });

        // Сохранение новых настроек
        buttonSaveIP.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        // Сохранение состояния флага тостов
        if (switchUseToast.isChecked()) {
            // Запись
            SharedPreferences prefs =
getSharedPreferences("AppPrefs", MODE_PRIVATE);
            prefs.edit().putBoolean("use_toast",
true).apply();
        } else {
            // Запись
            SharedPreferences prefs =
getSharedPreferences("AppPrefs", MODE_PRIVATE);
            prefs.edit().putBoolean("use_toast",
false).apply();
        }

        // Сохранение IP адресов устройств
        // Правильное получение текста с проверкой на
null
        String ip1 = editIP16BKF.getText() != null ?
            editIP16BKF.getText().toString() : "";
        String ip2 = editIPACerAspire.getText() != null
?
            editIPACerAspire.getText().toString() :
"";

        String configData = "ip1=" + ip1 + "\nip2=" +
ip2;

        Log.d("ConfigSave", configData);
        ConfigManager.saveConfig(SettingsActivity.this,
configData);

        showToast("Новые параметры сохранены");
    }
});

switchUseToast.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (switchUseToast.isChecked()) {
            Toast.makeText(SettingsActivity.this, "Тост-
сообщения включены", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(SettingsActivity.this, "Тост-
сообщения отключены", Toast.LENGTH_SHORT).show();
        }
    }
});
}

```

```
}
```

Файл «ControlActivity.java»

```
package com.example.commandertask;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.Socket;

public class ControlActivity extends AppCompatActivity {
    private static String SERVER_IP; // IP-адрес сервера
    private static int SERVER_PORT; // Port сервера
    private static String Name_Device;
    private EditText commandInput;
    private Button sendCommandButton;
    private Button increaseVolumeButton;
    private Button decreaseVolumeButton;
    private Button stopMediaButton;
    private Button noVolumeButton;
    private Button moveCursorUp, moveCursorDown, moveCursorLeft,
    moveCursorRight;
    private Button clickMouseLeft, clickMouseRight;
    private TextView textViewNameDevice;
    private ConstraintLayout rootView;
    private boolean flagUseToast;
    private SeekBar speedCursorBar;
    private SeekBar levelVolumeBar;
    private int speedCursor;
```

					ПГУ1.090301.16.001 ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

```

private int levelVolume;

private void showToast(String message) {
    if(!flagUseToast) return;
    Toast.makeText(this, message,
Toast.LENGTH_SHORT).show();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_control);

    // Чтение (в любом Activity)
    SharedPreferences prefs =
getSharedPreferences("AppPrefs", MODE_PRIVATE);
    flagUseToast = prefs.getBoolean("use_toast", false);

    // Получаем Intent, который запустил эту Activity
    Intent intent = getIntent();

    // Извлекаем данные по ключам
    SERVER_IP = intent.getStringExtra("IP_Address");
    SERVER_PORT = intent.getIntExtra("Port", 12345); //
12345 - значение по умолчанию
    Name_Device = intent.getStringExtra("Name_Device");

    // Получение имени ПК
    String command = "PC_NAME";
    new SendCommandTask().execute(command);

    commandInput = findViewById(R.id.commandInput);
    sendCommandButton =
findViewById(R.id.sendCommandButton);
    increaseVolumeButton = findViewById(R.id.button);
    decreaseVolumeButton = findViewById(R.id.button2);
    stopMediaButton = findViewById(R.id.button3);
    noVolumeButton = findViewById(R.id.button4);
    moveCursorUp = findViewById(R.id.button5);
    moveCursorDown = findViewById(R.id.button6);
    moveCursorLeft = findViewById(R.id.button7);
    moveCursorRight = findViewById(R.id.button8);
    clickMouseLeft = findViewById(R.id.button9);
    clickMouseRight = findViewById(R.id.button10);
    speedCursorBar = findViewById(R.id.seekBar);
    levelVolumeBar = findViewById(R.id.seekBar2);

    textViewNameDevice = findViewById(R.id.textView);
    textViewNameDevice.setText(Name_Device);

    rootView = findViewById(R.id.constraint2);

```

```

// Вешаем обработчик клика
rootView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Убираем фокус с EditText
        commandInput.clearFocus();

        // Скрываем клавиатуру
        InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
        if (imm != null) {

imm.hideSoftInputFromWindow(commandInput.getWindowToken(), 0);
        }
    }
});

speedCursorBar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SpeedBar speedBar, int
i, boolean b) {
        speedCursor = i;
        Log.d("speedCursor is:",
String.valueOf(speedCursor));
    }

    @Override
    public void onStartTrackingTouch(SpeedBar speedBar) {

    }

    @Override
    public void onStopTrackingTouch(SpeedBar speedBar) {

    }
});

levelVolumeBar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SpeedBar speedBar, int
i, boolean b) {
        levelVolume = i;
        Log.d("levelVolume is:",
String.valueOf(levelVolume));
    }

    @Override
    public void onStartTrackingTouch(SpeedBar speedBar) {

    }
}

```

					ПГУ1.090301.16.001 ПЗ	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

```

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
            String command = "SET_VOLUME " +
seekBar.getProgress();
            new SendCommandTask().execute(command);
            showToast("Установлена громкость " +
seekBar.getProgress());
        }
    });

    sendCommandButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String command =
commandInput.getText().toString();
            new SendCommandTask().execute(command);
        }
    });

    increaseVolumeButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String command = "INCREASE_VOLUME";
            new SendCommandTask().execute(command);

levelVolumeBar.setProgress(levelVolumeBar.getProgress()+2);
        }
    });

    decreaseVolumeButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String command = "DECREASE_VOLUME";
            new SendCommandTask().execute(command);

levelVolumeBar.setProgress(levelVolumeBar.getProgress()-2);
        }
    });

    stopMediaButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String command = "STOP_MEDIA";
            new SendCommandTask().execute(command);
        }
    });

```

```

        noVolumeButton.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        levelVolumeBar.setProgress(0);
        String command = "SET_VOLUME 0";
        new SendCommandTask().execute(command);
    }
});

        moveCursorUp.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String command = "MOVE_CUR_UP " + speedCursor;
        new SendCommandTask().execute(command);
    }
});

        moveCursorDown.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String command = "MOVE_CUR_DOWN " + speedCursor;
        new SendCommandTask().execute(command);
    }
});

        moveCursorLeft.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String command = "MOVE_CUR_LEFT " + speedCursor;
        new SendCommandTask().execute(command);
    }
});

        moveCursorRight.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String command = "MOVE_CUR_RIGHT " +
speedCursor;
        new SendCommandTask().execute(command);
    }
});

        clickMouseLeft.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String command = "CLICK_MOUSE_LEFT";
        new SendCommandTask().execute(command);
    }
}

```

```

    });
    clickMouseRight.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String command = "CLICK_MOUSE_RIGHT";
        new SendCommandTask().execute(command);
    }
});
}

private class SendCommandTask extends AsyncTask<String,
Void, String> {
    @Override
    protected String doInBackground(String... params) {
        String command = params[0];
        try (Socket socket = new Socket(SERVER_IP,
SERVER_PORT);
            DataOutputStream out = new
DataOutputStream(socket.getOutputStream());
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream())) {

            out.writeUTF(command);
            return in.readLine();
        } catch (Exception e) {
            Log.e("CommanderTask", "Error sending command",
e);
            return "Ошибка";
        }
    }

    @Override
    protected void onPostExecute(String result) {
        result = result.substring(2);

        if(result.startsWith("NAME")) {
            result = result.substring(4);
            Name_Device = result;
            textViewNameDevice.setText(Name_Device);
        }
        showToast("Ответ сервера: " + result);
    }
}

// Проверка разрешений
public boolean checkPermission(String permission, int
requestCode) {
    if
(ContextCompat.checkSelfPermission(ControlActivity.this,
permission) == PackageManager.PERMISSION_DENIED) {

```

```

ActivityCompat.requestPermissions(ControlActivity.this, new
String[]{permission}, requestCode);
        return false;
    } else {
        return true;
    }
}
}
}

```

Файл «ConfigManager.java»

```

package com.example.commandertask;

import android.content.Context;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class ConfigManager {
    private static final String CONFIG_FILE_NAME =
"app_config.cfg";

    // Сохранение конфигурации
    public static void saveConfig(Context context, String
configData) {
        try (FileOutputStream fos =
context.openFileOutput(CONFIG_FILE_NAME, Context.MODE_PRIVATE))
        {
            fos.write(configData.getBytes());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // Чтение конфигурации
    public static String loadConfig(Context context) {
        StringBuilder stringBuilder = new StringBuilder();
        try (FileInputStream fis =
context.openFileInput(CONFIG_FILE_NAME);
            InputStreamReader isr = new InputStreamReader(fis);
            BufferedReader br = new BufferedReader(isr)) {

            String line;
            while ((line = br.readLine()) != null) {
                stringBuilder.append(line).append("\n");
            }
        } catch (IOException e) {

```

```

        e.printStackTrace();
        return null;
    }
    return stringBuilder.toString();
}

// Получение пути к файлу (для ручного редактирования)
public static String getConfigFilePath(Context context) {
    return new File(context.getFilesDir(),
CONFIG_FILE_NAME).getAbsolutePath();
}
}

```

Файл «AddDeviceActivity.java»

```

package com.example.commandertask;

import android.os.Bundle;
import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;

public class AddDeviceActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_device);

        ImageView gifImageView =
findViewById(R.id.gifImageView);

Glide.with(this).load(R.drawable.loading).into(gifImageView);
    }
}

```

Файл «ServerForCommanderTask.java»

```

import java.awt.*;
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import com.sun.jna.*;
import com.sun.jna.platform.win32.WinDef;
import com.sun.jna.win32.StdCallLibrary;
import java.io.BufferedReader;

```

					ПГУ1.090301.16.001 ПЗ	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		

```

import java.io.InputStreamReader;

public class ServerForCommanderTask {
    private static final int SERVER_PORT = 12345;
    private static String NameDevice = "Неизвестное устройство";

    public static void main(String[] args) {
        try (ServerSocket serverSocket = new
ServerSocket(SERVER_PORT)) {
            NameDevice = GetPCModel();
            System.out.println("Модель ПК: " + NameDevice);
            System.out.println("Сервер прослушивает порт " +
SERVER_PORT);

            while (true) {
                new
ClientHandler(serverSocket.accept()).start();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    public static String GetPCModel() {
        try {
            Process process = Runtime.getRuntime().exec("wmic
computersystem get model, manufacturer");
            BufferedReader reader = new BufferedReader(new
InputStreamReader(process.getInputStream()));

            String line;
            while ((line = reader.readLine()) != null) {
                if (!line.trim().isEmpty() &&
!line.contains("Model") && !line.contains("Manufacturer")) {

```

					ПГУ1.090301.16.001 ПЗ	Лист
						73
Изм.	Лист	№ докум.	Подпись	Дата		

```

        return line.trim();
    }
}

} catch (Exception e) {
    e.printStackTrace();
}

return "Неизвестное устройство";
}

//Для паузы медиа
public interface User32Ext extends Library {
    User32Ext INSTANCE = Native.load("user32",
User32Ext.class);

    void keybd_event(byte bVk, byte bScan, int dwFlags, int
dwExtraInfo);
}

//для курсора
public interface User32 extends StdCallLibrary {
    User32 INSTANCE = Native.load("user32", User32.class);

    // Функция для перемещения курсора
    boolean SetCursorPos(int x, int y);

    // Функция для получения текущего положения курсора
    boolean GetCursorPos(WinDef.POINT point);
}

private static final byte VK_MEDIA_PLAY_PAUSE = (byte) 0xB3;
private static final byte VK_MEDIA_STOP = (byte) 0xB2;

private static final int KEYEVENTF_KEYUP = 0x0002;

private static void sendMediaKey(byte keyCode) {

```

					ПГУ1.090301.16.001 ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		

```

        User32Ext.INSTANCE.keybd_event(keyCode, (byte) 0, 0, 0);
        User32Ext.INSTANCE.keybd_event(keyCode,      (byte)      0,
KEYEVENTF_KEYUP, 0);
    }

    private static class ClientHandler extends Thread {
        private Socket socket;
        // Константы для событий мыши
        private static final int MOUSEEVENTF_LEFTDOWN = 0x0002;
        private static final int MOUSEEVENTF_LEFTUP = 0x0004;
        private static final int MOUSEEVENTF_RIGHTDOWN = 0x0008;
        private static final int MOUSEEVENTF_RIGHTUP = 0x0010;
        private static final int INPUT_MOUSE = 0;

        public ClientHandler(Socket socket) {
            this.socket = socket;
        }

        @Override
        public void run() {
            try          (DataInputStream      in          =          new
DataInputStream(socket.getInputStream()));
                DataOutputStream      out          =          new
DataOutputStream(socket.getOutputStream())) {

                String command = in.readUTF();
                System.out.println("Полученная команда: " +
command);

                if (command.startsWith("SEND_FILE")) {
                    receiveFile(command.substring(10).trim(),
in, out);
                } else {
                    String response = executeCommand(command);

```

```

        out.writeUTF(response);
    }
} catch (IOException ex) {
    ex.printStackTrace();
}
}

private void receiveFile(String fileName,
DataInputStream in, DataOutputStream out) throws IOException {
    File file = new File(fileName);
    try (FileOutputStream fileOut = new
FileOutputStream(file)) {
        byte[] buffer = new byte[4096];
        int bytesRead;
        while ((bytesRead = in.read(buffer)) != -1) {
            fileOut.write(buffer, 0, bytesRead);
        }
        out.writeUTF("Файл получен: " + fileName);
    }
}

private String executeCommand(String command) {
    try {
        if (command.equals("PC_NAME")) {
            return "NAME" + GetPCModel();

        } else if (command.startsWith("CMD")) {

Runtime.getRuntime().exec(command.substring(3));

            return "Выполнена команда " +
command.substring(3);

        } else if (command.equals("INCREASE_VOLUME")) {

```

```

        Runtime.getRuntime().exec("nircmd.exe
changesysvolume 1310");
        return "Громкость увеличена " + NameDevice;

    } else if (command.equals("DECREASE_VOLUME")) {
        Runtime.getRuntime().exec("nircmd.exe
changesysvolume -1310");
        return "Громкость уменьшена " + NameDevice;

    } else if (command.startsWith("SET_VOLUME")) {
        int                volume                =
Integer.parseInt(command.split(" ")[1]);
        Runtime.getRuntime().exec("nircmd.exe
setsysvolume " + 65535*volume/100);
        Toolkit.getDefaultToolkit().beep();      //
Системный звуковой сигнал
        return "Громкость " + NameDevice + "
установлена на " + volume;

    } else if (command.equals("STOP_MEDIA")) {
        //Runtime.getRuntime().exec("nircmd.exe
sendkeypress space");
        sendMediaKey(VK_MEDIA_PLAY_PAUSE);
        return "Медиа приостановлено " + NameDevice;

    } else if (command.startsWith("EXECUTE_FILE"))
{
        String                filePath                =
command.substring(12).trim();
        Runtime.getRuntime().exec(filePath);
        return "Запущен файл " + NameDevice + ": "
+ filePath;

    } else if (command.startsWith("LAUNCH_APP")) {

```

					ПГУ1.090301.16.001 ПЗ	Лист
						77
Изм.	Лист	№ докум.	Подпись	Дата		

```

        String appPath =
command.substring(11).trim();
        Runtime.getRuntime().exec(appPath);
        return "Запущено приложение " + NameDevice
+ ": " + appPath;

    } else if (command.startsWith("SHUTDOWN")) {
        int delay =
Integer.parseInt(command.split(" ")[1]);
        Runtime.getRuntime().exec("shutdown -s -t "
+ delay);
        return "Сервер " + NameDevice + " будет
выключен через " + delay + " секунд";

    } else if (command.equals("SHUTDOWN_NOW")) {
        Runtime.getRuntime().exec("shutdown -s -t
0");
        return "Сервер " + NameDevice + " будет
выключен прямо сейчас";

    } else if (command.equals("MOUSE_MOVE")) {
        // Implement mouse move command
        return "Мышь " + NameDevice + " перемещена";

    } else if (command.equals("KEYBOARD_TYPE")) {
        // Implement keyboard type command
        return "Включен режим клавиатуры " +
NameDevice;

    } else if (command.startsWith("MOVE_CUR_UP")) {
        int volume =
Integer.parseInt(command.split(" ")[1]);
        WinDef.POINT point = new WinDef.POINT();
        if (User32.INSTANCE.GetCursorPos(point)) {

```

					ПГУ1.090301.16.001 ПЗ	Лист
						78
Изм.	Лист	№ докум.	Подпись	Дата		

```

        int currentX = point.x;
        int currentY = point.y;
        int newY = currentY - volume;
        User32.INSTANCE.SetCursorPos(currentX,
newY);

    }

    return "Курсор " + NameDevice + " перемещен
вверх на " + Integer.toString(volume);

} else if (command.startsWith("MOVE_CUR_DOWN"))
{
    int volume =
Integer.parseInt(command.split(" ")[1]);
    WinDef.POINT point = new WinDef.POINT();
    if (User32.INSTANCE.GetCursorPos(point)) {
        int currentX = point.x;
        int currentY = point.y;
        int newY = currentY + volume;
        User32.INSTANCE.SetCursorPos(currentX,
newY);
    }

    return "Курсор " + NameDevice + " перемещен
вниз на " + Integer.toString(volume);

} else if (command.startsWith("MOVE_CUR_LEFT"))
{
    int volume =
Integer.parseInt(command.split(" ")[1]);
    WinDef.POINT point = new WinDef.POINT();
    if (User32.INSTANCE.GetCursorPos(point)) {
        int currentX = point.x;
        int currentY = point.y;
        int newX = currentX - volume;

```

```

        User32.INSTANCE.SetCursorPos(newX,
currentY);

    }

    return "Курсор " + NameDevice + " перемещен
влево на " + Integer.toString(volume);

        }                               else                               if
(command.startsWith("MOVE_CUR_RIGHT")) {
            int                               volume                               =
Integer.parseInt(command.split(" ")[1]);
            WinDef.POINT point = new WinDef.POINT();
            if (User32.INSTANCE.GetCursorPos(point)) {
                int currentX = point.x;
                int currentY = point.y;
                int newX = currentX + volume;
                User32.INSTANCE.SetCursorPos(newX,
currentY);
            }
            return "Курсор " + NameDevice + " перемещен
вправо на " + Integer.toString(volume);

        } else if (command.equals("CLICK_MOUSE_LEFT"))
{
            try {
                Runtime.getRuntime().exec("nircmd.exe
sendmouse left click");
            } catch (IOException e) {
                e.printStackTrace();
            }
            return NameDevice + " нажата левая кнопка
мышь";
        } else if (command.equals("CLICK_MOUSE_RIGHT"))
{
            try {

```

					ПГУ1.090301.16.001 ПЗ	Лист
						80
Изм.	Лист	№ докум.	Подпись	Дата		

```

        Runtime.getRuntime().exec("nircmd.exe
sendmouse right click");

        } catch (IOException e) {
            e.printStackTrace();
        }
        return NameDevice + " нажата левая кнопка
мышь";

    } else {
        return "Неизвестная команда на " +
NameDevice;
    }
} catch (Exception e) {
    e.printStackTrace();
    return "Ошибка выполнения команды на " +
NameDevice;
}
}
}
}

```

Файл «activity_add_device.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">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="@string/bg_home"
        android:scaleType="centerCrop"
        app:srcCompat="@drawable/main_background"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp" />
```

					<p style="text-align: center;"><i>ПГУ1.090301.16.001 ПЗ</i></p>	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

```

<ImageView
    android:id="@+id/imageView4"
    android:layout_width="210dp"
    android:layout_height="210dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/imageView2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.550"
    app:srcCompat="@drawable/pc_1500" />

```

```

<ImageView
    android:id="@+id/gifImageView"
    android:layout_width="81dp"
    android:layout_height="81dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.574"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.499"
    app:srcCompat="@drawable/loading" />

```

```

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Поиск новых устройств"
    android:textAlignment="center"
    android:textAllCaps="false"
    android:textColor="#FFFFFF"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.039" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

Файл «activity_control.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:id="@+id/constraint2"
    android:layout_width="wrap_content"

```

```

android:layout_height="match_parent"
android:backgroundTint="#00FFEB3B"
tools:context=".ControlActivity">

```

```

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="20dp" />

```

```

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:contentDescription="@string/bg_home"
    android:scaleType="centerCrop"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/main_background"
    tools:srcCompat="@drawable/main_background" />

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00000000"
    android:backgroundTint="#000000"
    android:orientation="vertical"
    android:padding="16dp"

```

```

    android:scrollbarThumbHorizontal="@drawable/main_background"
    tools:layout_editor_absoluteX="16dp"
    tools:layout_editor_absoluteY="0dp">

```

```

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="sans-serif-black"
    android:text="@string/device_num"
    android:textAlignment="center"
    android:textColor="#FFFFFF"
    android:textSize="24sp"
    android:textStyle="bold" />

```

```

<EditText
    android:id="@+id/commandInput"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="serif-monospace"

```

```

        android:hint="Введите команду"
        android:textColor="#FFFFFF"
        android:textColorHint="#FFFFFF"
        android:textSize="20sp" />

<Button
    android:id="@+id/sendCommandButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:backgroundTint="#80826E6E"
    android:text="Отправить команду CMD"
    android:textColor="#FFFFFF" />

<Space
    android:layout_width="match_parent"
    android:layout_height="20dp" />

<SeekBar
    android:id="@+id/seekBar2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="30" />

<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:backgroundTint="#80826E6E"
    android:text="Увеличить громкость"
    android:textColor="#FFFFFF" />

<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:backgroundTint="#80826E6E"
    android:text="Уменьшить громкость"
    android:textColor="#FFFFFF" />

<Button
    android:id="@+id/button3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:backgroundTint="#80826E6E"
    android:text="Пауза медиа"
    android:textColor="#FFFFFF" />

<Button
    android:id="@+id/button4"
    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:backgroundTint="#80826E6E"
        android:text="Без звука"
        android:textColor="#FFFFFF" />

<Space
    android:layout_width="match_parent"
    android:layout_height="20dp" />

<SeekBar
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="50" />

<Button
    android:id="@+id/button5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:backgroundTint="#80826E6E"
    android:text="Курсор мыши вверх"
    android:textColor="#FFFFFF" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/button7"
        android:layout_width="190dp"
        android:layout_height="wrap_content"
        android:backgroundTint="#80826E6E"
        android:text="Курсор мыши влево"
        android:textColor="#FFFFFF"
        android:textSize="13.5sp" />

    <Button
        android:id="@+id/button8"
        android:layout_width="190dp"
        android:layout_height="wrap_content"
        android:backgroundTint="#80826E6E"
        android:text="Курсор мыши вправо"
        android:textColor="#FFFFFF"
        android:textSize="13.5sp" />

</LinearLayout>

<Button
    android:id="@+id/button6"
    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:backgroundTint="#80826E6E"
        android:text="Курсор мыши вниз"
        android:textColor="#FFFFFF" />

<Space
    android:layout_width="match_parent"
    android:layout_height="20dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="57dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/button9"
        android:layout_width="190dp"
        android:layout_height="wrap_content"
        android:backgroundTint="#80826E6E"
        android:text="Нажать левую кнопку мыши"
        android:textColor="#FFFFFF" />

    <Button
        android:id="@+id/button10"
        android:layout_width="190dp"
        android:layout_height="wrap_content"
        android:backgroundTint="#80826E6E"
        android:text="Нажать правую кнопку мыши"
        android:textColor="#FFFFFF" />

</LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Файл «activity_devices.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:id="@+id/constraint"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="@string/bg_home"
    />

```

```

        android:scaleType="centerCrop"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/main_background"
        tools:srcCompat="@drawable/main_background" />

<ScrollView
    android:layout_width="128dp"
    android:layout_height="400dp"
    android:layout_marginStart="32dp"
    android:layout_marginTop="380dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <ImageButton
            android:id="@+id/imageButton"
            android:layout_width="128dp"
            android:layout_height="128dp"
            android:background="#00FFFFFF"
            android:contentDescription="@string/device_num"
            android:src="@drawable/pc_128"
            tools:src="@drawable/pc_128"
            tools:srcCompat="@drawable/pc_128" />

        <TextView
            android:id="@+id/textView2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/device_num"
            android:textAlignment="center"
            android:textColor="#FFFFFF"
            android:textSize="16sp" />

    </LinearLayout>
</ScrollView>

<ScrollView
    android:id="@+id/scrollView2"
    android:layout_width="128dp"
    android:layout_height="400dp"
    android:layout_marginTop="380dp"
    android:layout_marginEnd="32dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent">

```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <ImageButton
        android:id="@+id/imageButton2"
        android:layout_width="128dp"
        android:layout_height="128dp"
        android:background="#00FFFFFF"
        android:contentDescription="@string/device_num"
        android:src="@drawable/pc_128"
        tools:srcCompat="@drawable/pc_128" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/device_num"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textSize="16sp" />

</LinearLayout>
</ScrollView>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/textNameNetwork"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:selectAllOnFocus="false"
        android:singleLine="true"
        android:text="@string/name_network"
        android:textAlignment="center"
        android:textAllCaps="false"
        android:textColor="#FFFFFF"
        android:textSize="34sp"
        android:textStyle="bold" />
</com.google.android.material.textfield.TextInputLayout>

<ImageButton
    android:id="@+id/imageButton3"

```

```

android:layout_width="40dp"
android:layout_height="40dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
android:background="#00FFFFFF"
android:contentDescription="@string/settings"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/shestern_40"
tools:srcCompat="@drawable/shestern_40" />

```

```

<ImageButton
    android:id="@+id/imageButton4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="24dp"
    android:background="#00FFFFFF"
    android:contentDescription="@string/add_device"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/plus_40" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

Файл «activity_settings.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:id="@+id/con_settings"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

```

```

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:contentDescription="@string/bg_home"
    android:scaleType="centerCrop"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/main_background" />

```

```

<ImageView
    android:id="@+id/gifSettings2"
    android:layout_width="195dp"
    android:layout_height="189dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.726"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.241"
    app:srcCompat="@drawable/settings2" />

```

```

<ImageView
    android:id="@+id/gifSettings1"
    android:layout_width="250dp"
    android:layout_height="203dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.386"
    app:srcCompat="@drawable/settings1" />

```

```

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Идет настройка приложения"
    android:textAlignment="center"
    android:textColor="#FFFFFF"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.500"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.050" />

```

```

<Button
    android:id="@+id/SaveIP"
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:backgroundTint="#80826E6E"
    android:text="Сохранить"
    android:textColor="#FFFFFF"
    android:textColorLink="#80826E6E"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.502"

```

```

        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/textInputLayout3"
    app:layout_constraintVertical_bias="0.047" />

<Switch
    android:id="@+id/switchUseToast"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:checked="true"
    android:text="Использовать тосты"
    android:textAlignment="center"
    android:textColor="#FFFFFF"
    android:textColorLink="#80826E6E"
    android:textSize="16sp"

app:layout_constraintBottom_toTopOf="@+id/textInputLayout2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.971"
    tools:checked="true" />

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/textInputLayout2"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.502"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.629">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/ip16bkf"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autoText="true"
        android:hint="Введите IP 16BKF"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textSize="20sp" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/textInputLayout3"
    android:layout_width="200dp"
    android:layout_height="wrap_content"

```

```

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/textInputLayout2"
    app:layout_constraintVertical_bias="0.0">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/ipAcer"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Введите IP Acer Aspire"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textSize="20sp" />
    </com.google.android.material.textfield.TextInputLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Файл «colors.xml»

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFF</color>
</resources>

```

Файл «strings.xml»

```

<resources>
    <string name="app_name">CommanderTask</string>
    <string name="bg_home">Background Image Home</string>
    <string name="device_num">Device number </string>
    <string name="enter_the_nn">Enter the name your
network</string>
    <string name="settings">Settings</string>
    <string name="add_device">Add Device</string>
    <string name="name_network">Agency S</string>
</resources>

```

Файл «AndroidManifest.xml»

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission
android:name="android.permission.INTERNET"/>

```

					ПГУ1.090301.16.001 ПЗ	Лист
						92
Изм.	Лист	№ докум.	Подпись	Дата		

```

        <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
        <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.CommanderTask"
        tools:targetApi="31">
        <activity
            android:name=".DevicesActivity"
            android:windowSoftInputMode="adjustPan"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            </activity>
            <activity
                android:name=".ControlActivity"
                android:exported="true">
            </activity>
            <activity
                android:name=".SettingsActivity"
                android:exported="true">
            </activity>
            <activity
                android:name=".AddDeviceActivity"
                android:exported="true">
            </activity>
        </application>

</manifest>

```

Приложение В

(обязательное)

Приложение для удаленного управления ПК с мобильного устройства

Презентация

					ПГУ1.090301.16.001 ПЗ	Лист
						93
Изм.	Лист	№ докум.	Подпись	Дата		



Приложение для удаленного управления ПК с мобильного устройства

Автор работы: студент группы 21ВВП1 Мубаряков А.А.
Научный руководитель: к.т.н., доцент кафедры ВТ Юрова О.В.

Пенза, 2025

					ПГУ1.090301.16.001 ПЗ	Лист
						94
Изм.	Лист	№ докум.	Подпись	Дата		

Актуальность проекта

В наше время пользователи всё активнее стремятся к комфорту, автоматизации и удалённому доступу к своим цифровым устройствам. Одной из востребованных задач является возможность управления персональным компьютером (ПК) с мобильного устройства.

Это особенно актуально в ситуациях, когда требуется быстрое вмешательство — например, выключить ПК, управлять воспроизведением медиафайлов или эмулировать нажатия клавиш, находясь вне физического доступа к устройству.

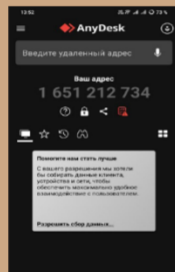
Цель проекта

Цель работы – разработать мобильное приложение, осуществляющее удаленное управление ПК по сети.

Мобильное приложение должно позволять пользователем просматривать список доступных к подключению устройств, выполнять безопасное подключение к ним, выполнять базовые функции по управлению ПК. Помимо отправки удаленных команд, необходимо реализовать управлением курсором и эмуляцию нажатия системных клавиш.

Обзор аналогов

AnyDesk



Достоинства:

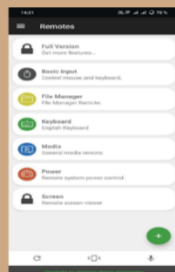
- Большой функционал.
- Приятное и понятное обучение.
- Возможность подключаться к устройствам по коду.

Недостатки:

- Тяжеловесная программа.
- Нет простоты.
- Требовательно к конфиденциальным данным.

Обзор аналогов

Unified Remote



Достоинства:

- Полный необходимый набор функций;
- Легкое приложение, понятный интерфейс;
- Простота настройки;
- Удобство использования.

Недостатки:

- Устаревший дизайн приложения;
- Высокая стоимость полной версии.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.16.001 ПЗ

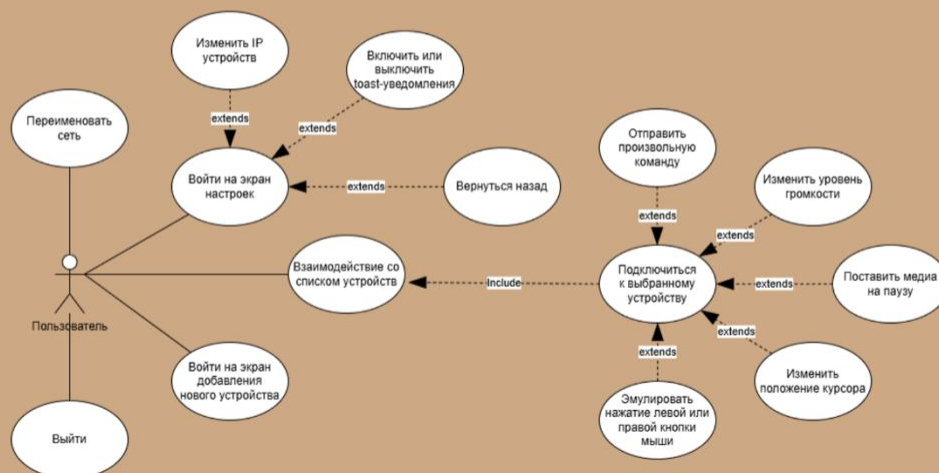
Лист

96

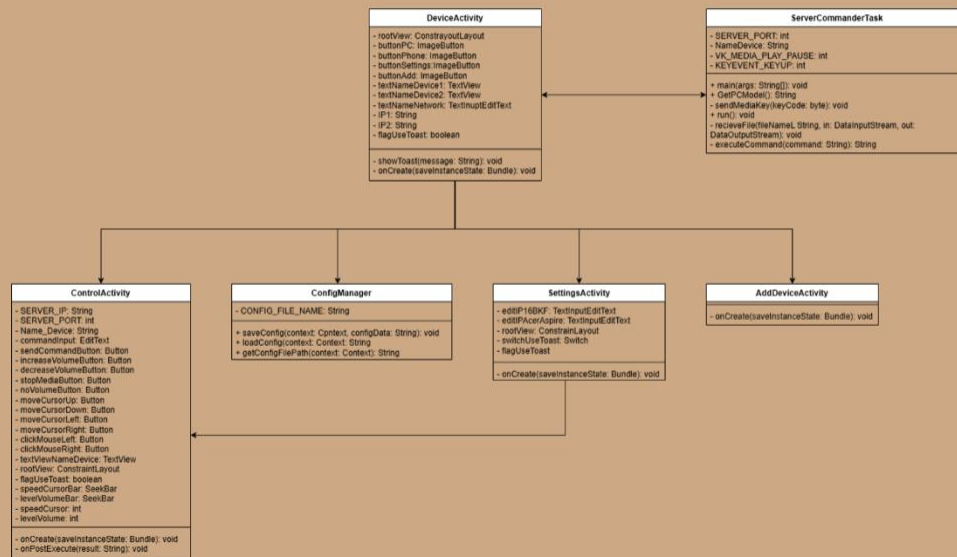
Стек технологий



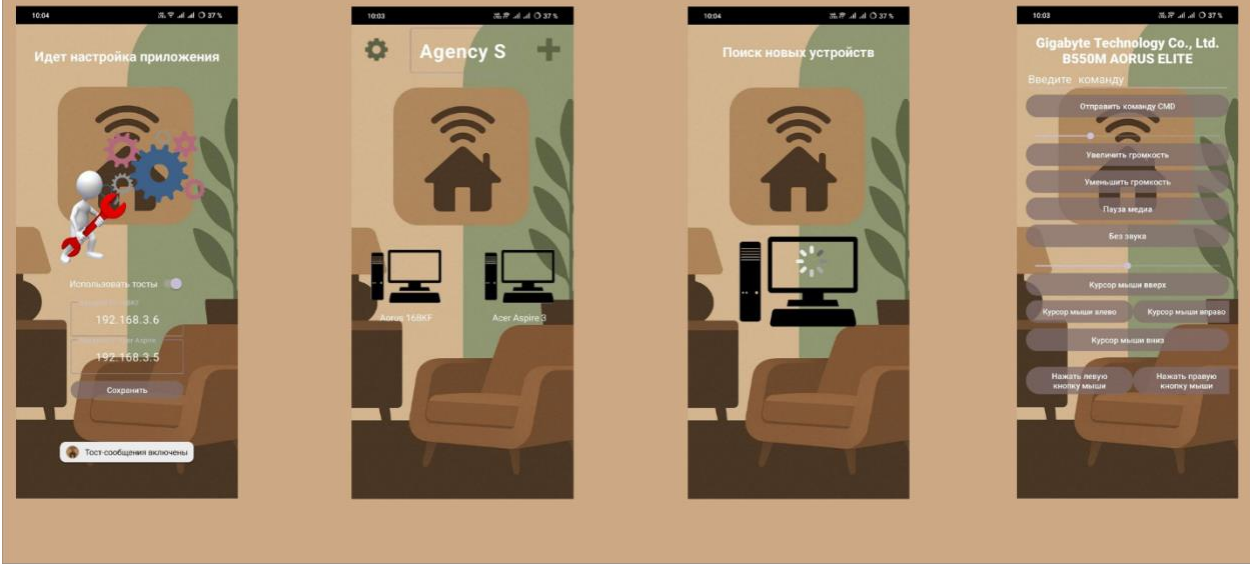
UML-Диаграмма вариантов использования



UML-Диаграмма классов



Внешний вид CommanderTask



Спасибо за внимание!