

Министерство науки и высшего образования Российской Федерации
Федеральное Государственное Образовательное Учреждение высшего
профессионального образования Национальный Исследовательский
Технологический Университет «МИСиС»
ИИТКН

Курсовая работа

по дисциплине «Современные инструментальные средства разработки»
«Проектирование и разработка мобильного приложения для организации
звонков»

Выполнили студенты МПИ-20-4-2:

Lead developer – Григорьев Константин

Database developer, business analyst – Дзюба Ирина

Autotest developer – Ионова Алена

System analyst, technical writer – Корлякова Екатерина

Frontend developer, project maintainer – Шилина Алена

Backend developer, technical writer – Юрасова Елена

Принял:

Тарханов И.А.

Москва, 2021

СОДЕРЖАНИЕ

Введение.....	3
1. Общие сведения о проекте	4
2. Функциональные требования к проекту	5
3. Нефункциональные требования	7
4. Сценарии использования.....	9
5. Сценарии тестирования	14
6. Прототипирование проекта.....	25
7. Основные API команды сервера.....	33
8. Модель базы данных.....	37
9. Результаты разработки проекта.....	40
Заключение	68
Список литературы	69
Приложение А. Листинг кода frontend-разработки.....	70
Приложение Б. Листинг кода backend-разработки.....	95
Приложение В. Листинг кода разработки БД и сервера.....	139

ВВЕДЕНИЕ

В настоящее время большую роль в жизни человека играют не массивные персональные компьютеры, а смартфоны и планшеты, которые уже могут заменить большинство функций компьютеров и кнопочных телефонов. Это неудивительно. Как известно, современные смартфоны производительнее и мощнее, чем компьютеры первых поколений, в том числе те, на которых разрабатывались космические программы.

Во всем мире самой популярной операционной системой для смартфонов остается Android (примерно 70% рынка), поэтому ориентация на пользователей Android предполагает большой охват пользователей и большую доступность приложения. Также традиционно приложения под Android достаточно легко выставить в общий доступ. Это и площадка PlayMarket, и установка приложений из сторонних источников, что позволяет разработчикам получать доступ к своей целевой аудитории.

Сегмент мобильных приложений растет в соответствии с требованиями времени и спросом. Пользователям необходимы удобные инструменты работы с документами, способы общения с коллегами, развлечения и планировщики.

В основе идеи проекта лежит создание приложения для организации входящих вызовов и планирования звонков – согласование переноса разговора на подходящее время. Наиболее ярким примером применения приложения служит автоматический перенос звонка на ближайшее свободное время при входящем вызове во время важного мероприятия.

1. ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТЕ

1.1 Название проекта

Полное название проекта: ProjectX

Условные обозначения: Проект, Приложение, Система.

1.2 Исполнители

Lead developer – Григорьев Константин

Database developer, business analyst – Дзюба Ирина

Autotest developer – Ионова Алена

System analyst, technical writer – Корлякова Екатерина

Frontend developer, project maintainer – Шилина Алена

Backend developer, technical writer – Юрасова Елена

1.3 Цель проекта

Создать мобильное приложение на базе Android для организации и переноса звонков на основе мероприятий в календаре мобильного телефона.

1.4 Задачи проекта

1. Возможность переноса звонков пользователя на удобное время.
2. Возможность планирования звонков.
3. Отправка автоматических сообщений от пользователя на нужный номер с временем для перенесенного звонка.
4. Выбор конфигурации настроек приложения: режимов и принципов работы.
5. Синхронизация приложения на устройствах пользователя.
6. Доступ к дополнительным функциям при покупке премиум-аккаунта.

2. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОЕКТУ

2.1. Принятие разрешений

Для корректной работы приложения необходимо получение специфических разрешений, например приема звонков. Прежде чем перейти к регистрации при первом запуске пользователь попадает на экран со списком требуемых разрешений, где может принять их или изучить. Если часть разрешений не принята, система уведомляет об этом и просит принять.

2.2. Регистрация и авторизация

Использование возможностей приложения должно быть возможно только после идентификации пользователя. Процедура регистрации должна быть реализована в отдельной форме и привязана к аккаунту Google (почта gmail). Процедура авторизации возможна так же по аккаунту Google. При возникновении ошибки всплывает соответствующее уведомление.

2.3. Премиум-аккаунт

Пользователю может быть предложен доступ к дополнительным функциям приложения при внесении оплаты и покупки премиум-аккаунта. Система предлагает форму оплаты, где авторизованный пользователь может онлайн-платежом перевести деньги приложению, после чего его аккаунт становится «премиум», а пользователь получает доступ к функциям.

2.4. Настройки дизайна

Пользователь с премиум-аккаунтом получает возможность выбрать одну из трех реализаций цветового решения приложения и соответствующего дизайна. В настройках приложения в разделе «оформление» пользователь может отметить выбранный дизайн, посмотреть, как будет выглядеть работающее приложение и принять изменения, после чего система будет использовать этот вариант оформления.

2.5. Настройки времени

В настройках приложения в разделе «время» пользователь определяет один из нескольких вариантов переноса звонков: автоматическое – на ближайшее в календаре свободное время; ручное– предлагать и выбор времени, и вариант свободного времени, и вариант переноса на определенное количество минут (полчаса/час/...).

2.6. Настройки использования

В настройках приложения пользователь так же выбирает, должно ли приложение реагировать на все входящие вызовы (только для ручного режима) или только во время мероприятий (для ручного и автоматического режима).

2.7. Синхронизация

Данные о переносе и планировании звонков автоматически синхронизируются на всех устройствах, где авторизован пользователь.

2.8. Функция звонка

При поступлении входящего вызова всплывает иконка приложения. При нажатии на нее всплывает форма выбора времени с вариантами ручной настройки, автоматического переноса на свободное время в календаре, переноса звонка на час/полчаса вперед. Когда пользователь выбирает один из вариантов (или в случае, если в настройках была выбрана автоматическая отправка, в таком случае участие пользователя не требуется), система завершает входящий вызов, отправляет звонившему сообщение с просьбой перезвонить в определенное время и отмечает в календаре время предстоящего звонка.

2.9. Функция планировщика звонков

При выборе данной функции в приложении пользователю предлагается форма для выбора номера телефона из контактов или его введения вручную, форма выбора времени. После подтверждения выбора система отправляет по указанному номеру сообщение с просьбой созвониться в определенное время, а в календаре отмечается время предстоящего звонка.

3. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

3.1. Требования к пользователям приложения

Для работы с приложением от пользователей не должно требоваться технических навыков и знаний, за исключением общих навыков работы с приложениями на смартфонах.

Для регистрации и авторизации пользователей необходима почта аккаунта Google.

3.2. Требования к надежности

Программный комплекс должен обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие аварийные сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

3.3. Требования к производительности

Время отклика системы на запрос пользователя не должно превышать 2 секунд.

Связь с сервером должна осуществляться при авторизации и изменении настроек приложения, соответственно, допустимое количество одновременных запросов к серверу может быть ограничено 50 человек.

3.4. Требование к безопасности

Вся личная информация (пользовательские данные, включая доступ к аккаунту Google, номер телефона, пользовательские настройки) не должны быть доступны сторонним пользователям.

3.5. Требования к эргономике и технической эстетике

Взаимодействие пользователей с программным комплексом должно осуществляться посредством визуального графического интерфейса (GUI). Ввод-вывод данных, прием управляющих команд и отображение результатов

их исполнения должны выполняться в интерактивном режиме, в реальном режиме времени. Интерфейс должен быть рассчитан на преимущественное использование мобильных телефонов (смартфонов) с разрешением 1080x2220: 440dpi. Управление осуществляется с помощью набора экранных кнопок, меню и форм выбора. При вводе и редактировании текстовых данных должна использоваться экранная клавиатура мобильного телефона.

3.6. Требования к лингвистическому обеспечению

Пользователи должны взаимодействовать с системой на уровне графического пользовательского интерфейса.

Все функции системы должны поддерживать русский язык.

3.7. Требования к техническому обеспечению

Система должна быть рассчитана на функционирование в следующей программной среде: мобильные устройства на базе Android.

При реализации серверной и клиентской частей системы должны применяться следующие

языки высокого уровня и библиотеки:

Базы данных: MySQL 8.0.19 (Open Server Panel 5.3.7)

Сервер: PHP 8.0.01 (Open Server Panel 5.3.7)

Back: JAVA 1.6 (Android Studio 4.1.2)

Front: XML 1.0, JAVA 1.6 (Android Studio 4.1.2)

Для работы сервера приложения выполняются следующие требования:

Локальный веб-сервер установлен на ОС Windows;

Наличие 64-битной системы;

Не меньше 500 МБ RAM;

Не меньше 4 ГБ HDD.

4. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

Сценарии использования или юзкейсы (Use Case) описывают поведение системы при взаимодействии с ней пользователя. Они дополняют и раскрывают функциональные требования к системе и описываются определенным шаблоном. Уровень детализации шаблона зависит от сложности системы и ее задач. Шаблон обязательно должен включать в себя название и цель сценария использования; акторов или действующие лица, то есть того, кто взаимодействует с системой; порядок событий или сценарий; возможно возникновение альтернативного сценария, но приветствуется минимизация числа условий и развилок внутри сценария. Так же применяются активаторы и предварительные условия, определяющие возможность наступления того или иного сценария.

4.1 Use Case «Получение разрешений»

Необходима при первом входе в приложение

Цель: приложение должно получить разрешения, регулирующие его возможности

Действующее лицо: Пользователь приложения (П).

Основной сценарий:

1. П заходит в приложение впервые (или есть еще не принятые разрешения).
2. Система предлагает кнопку «принять все» и список разрешений с выделенными не принятыми разрешениями
3. П может получить информацию о каждом из разрешений и принять его.
4. П получает доступ к дальнейшим действиям в приложении.

Альтернативные сценарии:

2.a П выбрал «принять все».

2.a.1 Система автоматически проставляет все разрешения.

2.a.2 На экране смартфона выводится информация о всех принятых разрешениях (на несколько секунд или с кнопкой «понятно»

2.a.3 П получает доступ к дальнейшим действиям в приложении.

3.a П пропускает какое-то разрешение и выходит.

3.a.1 При следующем запуске система вновь предлагает разрешения.

4.2 Use Case «Авторизация пользователя»

Необходима при первом входе или при выходе из аккаунта.

Цель: дать возможность пользователю зарегистрироваться в системе и войти в нее.

Действующее лицо: Пользователь приложения (П).

Основной сценарий:

1. П заходит в приложение.
2. Если авторизация не была выполнена ранее (система проверяет это), предлагается выполнить вход или зарегистрироваться.
3. П выполняет вход – подтверждает свою личность установленным способом (почта gmail).
4. П получает доступ к функционалу системы и сохраненным настройкам использования.

Альтернативные сценарии:

2.a П выбрал регистрацию.

2.a.1 Предлагается зарегистрироваться с использованием почты gmail google-аккаунта.

2.a.2 Регистрация подтверждается, выполняется вход.

2.a.3 П получает доступ к функционалу системы и сохраненным настройкам использования.

2.b П уже авторизован, получает доступ к функционалу системы.

2.b.1 П получает доступ к функционалу системы и сохраненным настройкам использования.

4.3 Use Case «Покупка премиум-аккаунта»

Предлагается пользователю оплата за дополнительные функции

Цель: монетизация кастомизации приложения (смены дизайна иконки и цвета)

Действующее лицо: Пользователь приложения без премиум-аккаунта (Пб).

Основной сценарий:

1. Пб заходит в раздел платных функций (настроек).
2. Пб при выборе функции получает краткую информацию о ней.
3. Система информирует Пб о недоступности функции для его аккаунта.
4. Система предлагает купить премиум-аккаунт за установленную цену.

5. Пб соглашается и переходит на форму оплаты.
6. После завершения оплаты Пб получает премиум-аккаунт и доступ к платным функциям.

Альтернативные сценарии:

4.a Пб отказался от предложения или отменил его в процессе оплаты.

4.a.1 Система возвращается к разделу платных функций.

4.4 Use Case «Настройка дизайна»

Доступна для авторизованных пользователей, кастомизация приложения.

Цель: дать возможность пользователю с премиум-аккаунтом изменить цветовую схему и элементы дизайна.

Действующее лицо: Пользователь приложения с премиум-аккаунтом (Пп)

Основной сценарий:

1. Пп заходит в настройки приложения, в кастомизацию.
2. Пп выбирает один из вариантов изменения иконки и цветовой схемы (акула/енот/пантера).
3. Система окрашивается согласно выбору, иконка демонстрируется Пп.
4. Цветовая схема и иконка приложения меняются на заданную.

4.5 Use Case «Настройка работы вне мероприятия»

Доступна для авторизованных пользователей, изменяет алгоритм действий при звонке.

Цель: дать возможность пользователю выбрать алгоритм, по которому будет действовать приложение во время звонка, если в календаре это время отмечено как свободное.

Действующее лицо: Пользователь приложения (независимо от того, премиум или обычный) (П)

Основной сценарий:

1. П заходит в настройки приложения, в раздел времени.

2. П выбирает вариант работы алгоритма (только вручную или не использовать приложение)
3. П подтверждает или отклоняет выбор.
4. Алгоритм выбирается системой как основной для звонков вне мероприятий.

4.6 Use Case «Настройка работы на мероприятиях»

Доступна для авторизованных пользователей, изменяет алгоритм действий при звонке.

Цель: дать возможность пользователю выбрать алгоритм, по которому будет действовать приложение во время звонка на мероприятии.

Действующее лицо: Пользователь приложения (независимо от того, премиум или обычный) (П)

Основной сценарий:

1. П заходит в настройки приложения, в раздел времени.
2. П выбирает один из вариантов работы алгоритма (автоматически, только вручную)
3. П подтверждает или отклоняет выбор.
4. Алгоритм выбирается системой как основной для звонков во время мероприятий

4.7 Use Case «Звонок»

Основная функция приложения.

Цель: перенос звонков во время мероприятия на свободное время.

Действующее лицо: Пользователь приложения (П).

Основной сценарий:

1. Поступает входящий вызов.
2. Если в настройках работы текущее время соответствует разрешенному (выбор между «всегда» и «во время мероприятия» UC5), то приложение начинает функционировать в активном режиме.
3. Если не выбран автоматический перенос в настройках, всплывает иконка.
4. Всплывающая иконка предлагает П воспользоваться приложением.
5. П устанавливает время или выбирает предлагаемое (ближайшее свободное время в календаре, через час/etc)
6. П подтверждает выбор.
7. Звонок сбрасывается.

8. Отправляется смс звонящему с выбранным временем и просьбой перезвонить.
9. В календаре отмечается время возможного звонка.
10. На других устройствах пользователя так же появляется информация о возможном звонке.

Альтернативные сценарии:

2.a Звонок поступил не во время мероприятия при выборе варианта активной работы только во время мероприятий.

2.a.1 Приложение функционирует в фоновом режиме.

3.a Если выбран автоматический перенос во время мероприятия.

3.a.1 Звонок сбрасывается.

3.a.2 Выбирается ближайшее свободное время в календаре.

3.a.3 Отправляется смс звонящему с этим временем и просьбой перезвонить.

3.a.4 В календаре отмечается время возможного звонка.

3.a.5 На других устройствах пользователя так же появляется информация о возможном звонке.

4.a П отклоняет предложение использования приложения.

4.a.1 Приложение переходит в фоновый режим.

4.8 Use Case «Планировщик»

Основная функция приложения.

Цель: выбор времени созвона с другим человеком.

Действующее лицо: Пользователь приложения (П)

Основной сценарий:

1. П выбирает «Планировщик звонков»
2. П вводит номер телефона или выбирает из списка контактов.
3. П устанавливает время или выбирает предлагаемое (ближайшее свободное время в календаре, через час/etc)
4. П подтверждает выбор.
5. Отправляется смс по номеру с выбранным временем и предложением созвониться.
6. В календаре отмечается время предстоящего звонка.
7. На других устройствах пользователя так же появляется информация о возможном звонке.

5. СЦЕНАРИИ ТЕСТИРОВАНИЯ

Сценарии тестирования или тесткейсы (Test Case) – это профессиональная документация тестировщика, последовательность действий, направленная на проверку какого-либо функционала, описывающая как достичь фактического результата. Разработка тесткейсов основана на сценариях использования и функциональных и нефункциональных требованиях к системе, например, проверка поддержки русского языка во всем приложении. Шаблон тесткейса всегда включает в себя название; уникальный идентификатор, необходимый для организации работы тестировщика; предусловие; шаги тестирования, то есть, описание последовательности действий и ожидаемого результата. Для каждого из тесткейсов можно получить положительный результат, совпадающий с ожидаемым; отрицательный, когда получен иной результат, что означает наличие ошибки; невозможность выполнения теста, что так же означает наличие ошибки в системе.

5.1 Test Case «Принятие разрешений вручную после установки»

ID 001

Предусловия

Установить приложение

Шаги

№	Действие	Ожидаемый результат
1	Запустить приложение в первый раз	Отобразился список разрешений с выделенными непринятыми разрешениями: <ul style="list-style-type: none">• Системные разрешения• Включать поверх приложений• Разрешить в фоновом режиме• Отключить энергосбережение
2	Получить информацию о каждом разрешении	Информация отобразилась
3	Принять каждое не принятое разрешение	Получен доступ к приложению

5.2 Test Case «Принятие не всех разрешений»

ID 002

Предусловия

Установить приложение

Шаги

№	Действие	Ожидаемый результат
1	Запустить приложение первый раз	Отобразился список разрешений с выделенными непринятыми разрешениями: <ul style="list-style-type: none">• Системные разрешения• Включать поверх приложений• Разрешить в фоновом режиме• Отключить энергосбережение
2	Принять разрешение А Пропустить разрешение Б Нажать «Продолжить»	Отобразился алерт «Еще не все разрешения приняты». Доступ к приложению не был предоставлен

5.3 Test Case «Успешная авторизация пользователя»

ID 101

Предусловия

Установить приложение, принять все разрешения

Шаги

№	Действие	Ожидаемый результат
1	Запустить приложение	Отображено окно с выбором «Войти/Зарегистрироваться»
2	Ввести верные данные для авторизации Нажать «Войти»	Авторизация пройдена успешно Получен доступ к приложению

5.4 Test Case «Неудачная авторизация пользователя. email»

ID 102

Предусловия

Установить приложение, принять все разрешения

Шаги

№	Действие	Ожидаемый результат
1	Запустить приложение	Отображено окно с выбором «Войти/Зарегистрироваться»
2	Ввести неверные данные для авторизации (неверный email) Нажать «Войти»	Авторизация не пройдена Отобразился алерт «Ошибка авторизации: Пользователь с такой почтой не найден» Доступ к приложению не предоставлен

5.5 Test Case «Неудачная авторизация пользователя. пароль»

ID 103

Предусловия

Установить приложение, принять все разрешения

Шаги

№	Действие	Ожидаемый результат
1	Запустить приложение	Отображено окно с выбором «Войти/Зарегистрироваться»
2	Ввести неверные данные для авторизации (неверный пароль) Нажать «Войти»	Авторизация не пройдена Отобразился алерт «Ошибка авторизации: Неверно введен пароль» Доступ к приложению не предоставлен

5.6 Test Case «Покупка премиум-аккаунта»

ID 201

Предусловия

Установить приложение, принять все разрешения, авторизоваться

Шаги

№	Действие	Ожидаемый результат
1	Зайти в раздел «Мой статус»	Отображена информация о том, что статус стандартный, и кнопка «Купить премиум»
2	Согласиться на покупку премиум-аккаунта	Переход на форму оплаты
3	Оплатить премиум-аккаунт	Премиум-аккаунт куплен Доступ к смене цветовой темы предоставлен

5.7 Test Case «Успешная смена цветовой схемы»

ID 301

Предусловия

Установить приложение, принять все разрешения, авторизоваться,
купить премиум-аккаунт

Шаги

№	Действие	Ожидаемый результат
1	Зайти в раздел «Настройки»	Отображен выпадающий список тем: <ul style="list-style-type: none">• Акула• Енот• Пантера
2	Выбрать вариант кастомизации	Система окрасилась в соответствии с выбранной схемой, иконка отобразилась пользователю
4	Выйти из приложения Зайти в приложение	Цветовая схема и иконка не изменились

5.8 Test Case «Неудачная смена цветовой схемы»

ID 302

Предусловия

Установить приложение, принять все разрешения, авторизоваться

Шаги

№	Действие	Ожидаемый результат
1	Зайти в раздел «Настройки»	Отображен выпадающий список тем: <ul style="list-style-type: none">• Акула• Енот• Пантера
2	Выбрать вариант кастомизации	Отобразился алерт «Чтобы сменить тему, купите премиум»

5.9 Test Case «Настройка работы алгоритма и варианта использования»

ID 4501

Предусловия

Установить приложение, принять все разрешения

Шаги

№	Действие	Ожидаемый результат
1	Зайти в настройки работы алгоритма	Отображены варианты работы алгоритма для режима «во время мероприятий» <ul style="list-style-type: none">• Вручную (выбран по умолчанию)• Авто (на свободное время) Отображены варианты работы алгоритма для режима «вне мероприятий» <ul style="list-style-type: none">• Вручную (выбран по умолчанию)• • Выключить
2	Сменить вариант работы алгоритма Выйти из настроек Зайти в настройки	Выбран сохраненный вариант работы алгоритма

5.10 Test Case «Неудачная настройка работы алгоритма и варианта использования»

ID 4502

Предусловия

Установить приложение, принять все разрешения

Шаги

№	Действие	Ожидаемый результат
1	Зайти в настройки работы алгоритма	Отображены варианты работы алгоритма для режима «во время мероприятий» <ul style="list-style-type: none">• Вручную (выбран по умолчанию)• Авто (на свободное время) Отображены варианты работы алгоритма для режима «вне мероприятий» <ul style="list-style-type: none">• Вручную (выбран по умолчанию)• Выключить
2	Сменить настройки работы алгоритма Нажать кнопку «Сбросить настройки»	Настройки возвращены к значению по умолчанию

5.11 Test Case «Перенос звонка во время мероприятий (автоматический)»

ID 701

Предусловия

Установить приложение, принять все разрешения, авторизоваться, выбрать вариант работы алгоритма «автоматический» для режима «во время мероприятий»

Шаги

№	Действие	Ожидаемый результат
1	С устройства А установить встречу	Встреча сохранена
2	С устройства Б позвонить на устройство А во время встречи	На устройстве А создалась встреча звонка на время после окончания встречи длительностью 5 минут На устройстве А в приложении в разделе «Отложенные звонки» отобразился перенесенный звонок Устройство Б оповещено о встрече устройства А, времени завершения встречи и создания встречи после нее

5.12 Test Case «Успешный перенос звонка во время мероприятий (ручной)»

ID 702

Предусловия

Установить приложение, принять все разрешения, авторизоваться, выбрать вариант работы алгоритма «ручной»

Шаги

№	Действие	Ожидаемый результат
1	С устройства А установить встречу в календаре	Встреча сохранена
2	С устройства Б позвонить на устройство А во время встречи	На устройстве А отображается окно для принятия/отклонения звонка и иконка приложения
3	Нажать на иконку приложения	Отобразилось окно для выбора времени начала встречи (для отложенного звонка)
4	Выбрать время для отложенного звонка после завершения текущей встречи	Встреча создана в соответствии с выбранным временем длительностью 5 минут На устройстве А в приложении в разделе «Отложенные звонки» отобразился перенесенный звонок Устройство Б оповещено о текущей встрече устройства А, времени ее завершения и создания встречи после неё

5.13 Test Case «Неудачный перенос звонка во время мероприятий (ручной)»

ID 703

Предусловия

Установить приложение, принять все разрешения, авторизоваться, выбрать вариант работы алгоритма «ручной»

Шаги

№	Действие	Ожидаемый результат
1	С устройства А установить встречу в календаре	Встреча сохранена
2	С устройства Б позвонить на устройство А во время встречи	На устройстве А отображается окно для принятия/отклонения звонка и иконка приложения
3	Нажать на иконку приложения	Отобразилось окно для выбора времени начала встречи (для отложенного звонка)
4	Выбрать время для отложенного звонка до завершения текущей встречи	Отобразился алерт «На выбранное время уже запланировано другое мероприятие»

5.14 Test Case «Принятие звонка во время мероприятий»

ID 703

Предусловия

Установить приложение, принять все разрешения, авторизоваться, выбрать вариант работы алгоритма «ручной»

Шаги

№	Действие	Ожидаемый результат
1	С устройства А установить встречу в календаре	Встреча сохранена
2	С устройства Б позвонить на устройство А во время встречи	На устройстве А отображается окно для принятия/отклонения звонка и иконка приложения
3	Выбрать вариант действия «принять звонок»	Звонок принят

5.15 Test Case «Перенос звонка вне мероприятий (ручной)»

ID 705

Предусловия

Установить приложение, принять все разрешения, авторизоваться, выбрать для режима «вне мероприятий» вариант работы алгоритма «ручной»

Шаги

№	Действие	Ожидаемый результат
1	С устройства Б позвонить на устройство А во время встречи	На устройстве А отображается окно для принятия/отклонения звонка и иконка приложения
2	Нажать на иконку приложения	Отобразилось окно для выбора времени начала встречи (для отложенного звонка)
3	Выбрать время для отложенного звонка	Встреча создана в соответствии с выбранным временем длительностью 5 минут На устройстве А в приложении в разделе «Отложенные звонки» отобразился перенесенный звонок Устройство Б оповещено о текущей встрече устройства А, времени ее завершения и создания встречи после неё

5.16 Test Case «Перенос звонка вне мероприятий (выключен)»

ID 705

Предусловия

Установить приложение, принять все разрешения, авторизоваться, выбрать для режима «вне мероприятий» вариант работы алгоритма «выключен»

Шаги

№	Действие	Ожидаемый результат
1	С устройства Б позвонить на устройство А	Звонок не был перенесен, иконка приложения не отобразилась

5.17 Test Case «Успешное планирование звонка»

ID 801

Предусловия

Установить приложение на устройство А, принять все разрешения, авторизоваться

Шаги

№	Действие	Ожидаемый результат
1	Зайти в «планировщик звонков» с устройства А Ввести номер телефона устройства Б Выбрать свободное время Подтвердить выбор	На устройство Б пришло СМС с выбранным временем и предложением созвониться В календаре устройства А отмечено время предстоящего звонка длительностью 5 минут

5.18 Test Case «Неудачное планирование звонка»

ID 802

Предусловия

Установить приложение на устройство А, принять все разрешения, авторизоваться

Шаги

№	Действие	Ожидаемый результат
1	Зайти в «планировщик звонков» с устройства А Ввести номер телефона устройства Б Выбрать время, занятое другой встречей Подтвердить выбор	Отобразился алерт «На выбранное время уже запланировано другое мероприятие»

6. ПРОТОТИПИРОВАНИЕ ПРОЕКТА

Создание прототипа приложения позволяет оценить предполагаемую реализацию до проведения основных работ по разработке.

На этапе первоначального прототипирования так же легко графически обозначить связи между различными элементами пользовательского интерфейса, рисунок 6.1, например, переход после авторизации пользователя на основную страницу приложения.

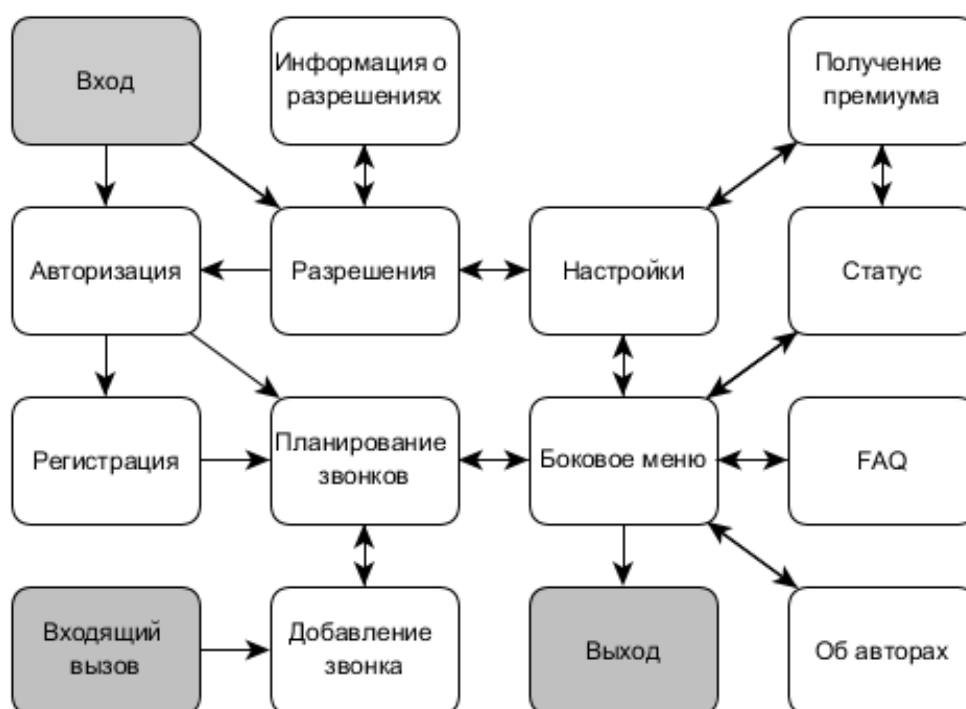


Рис. 6.1 – Схема структуры приложения

Быстрое прототипирование, то есть, создание макетов интерфейса, которые не войдут потом в основную программу, позволяет внести изменения в концепцию проекта на ранней стадии, что минимизирует трудозатраты разработчиков. Эволюционное прототипирование заключается в постепенном усложнении первого прототипа до конечного пользовательского интерфейса.

На рисунках 6.2-6.13 первые прототипы экранов приложения. На рисунках 6.14-6.21 усложненный вариант прототипирования с использованием основной цветовой темы приложения.



Рис. 6.2 – Разрешения

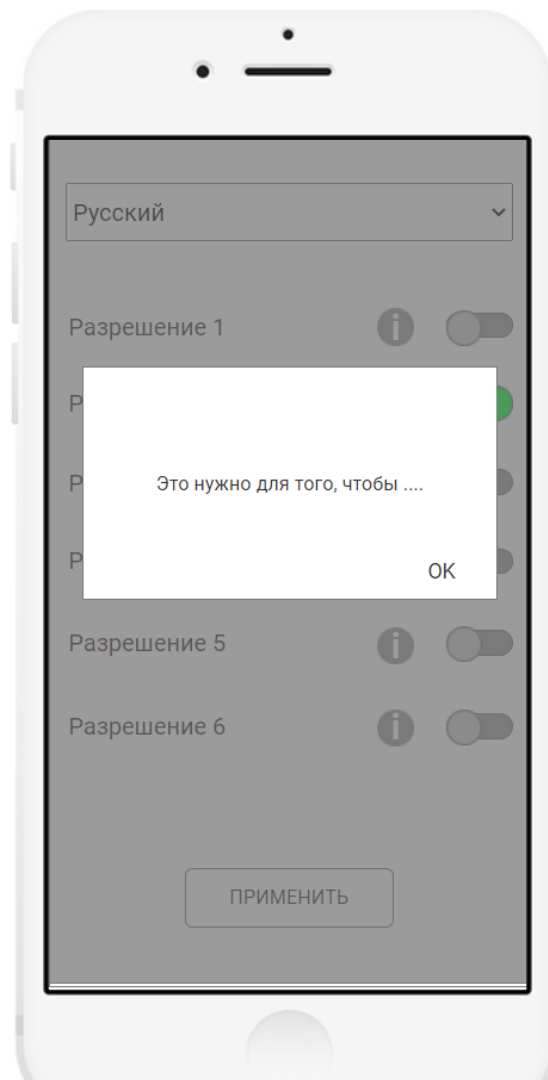


Рис. 6.3 – Информация о разрешениях

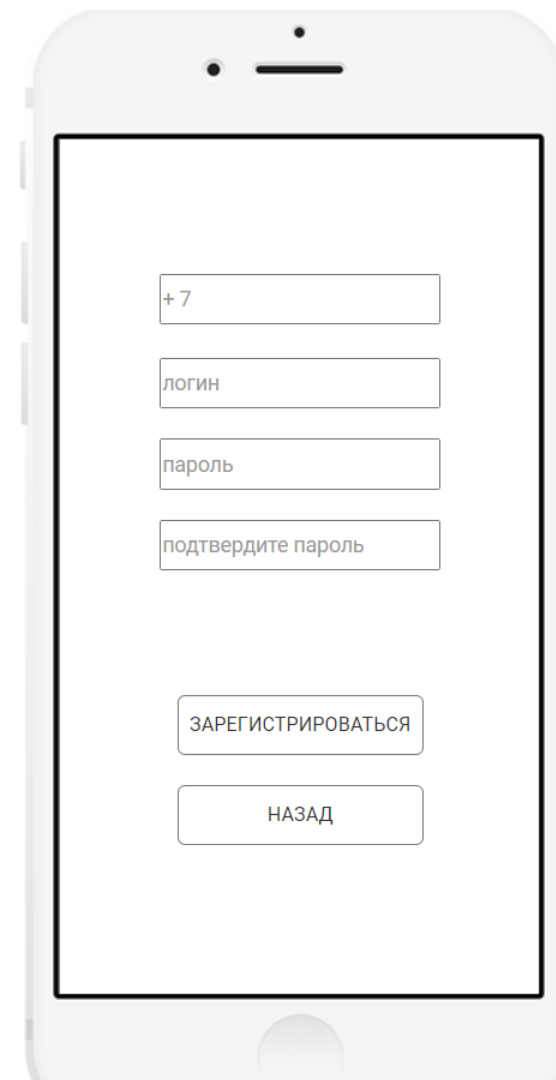


Рис. 6.4 – Регистрация

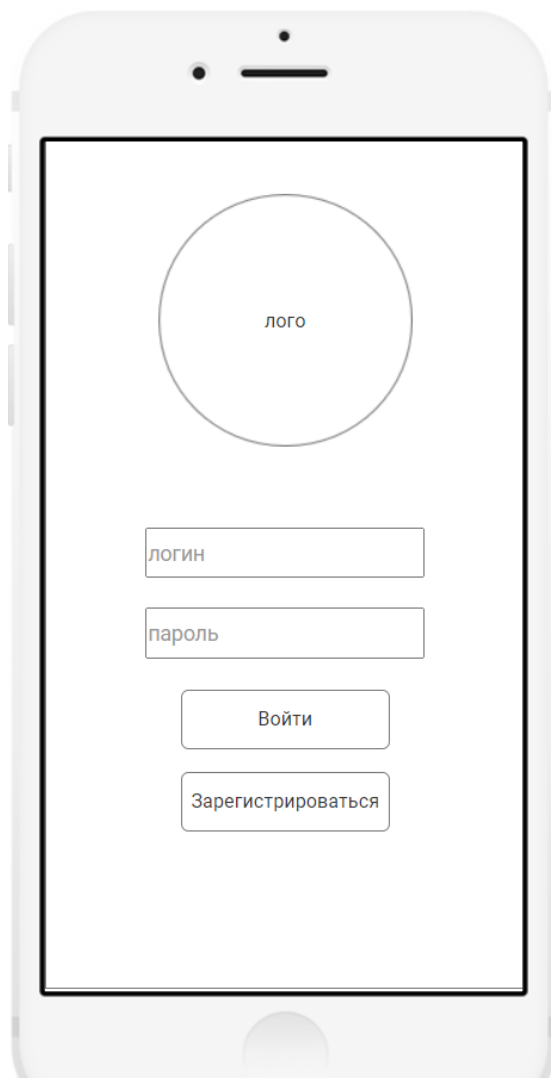


Рис. 6.5 – Авторизация

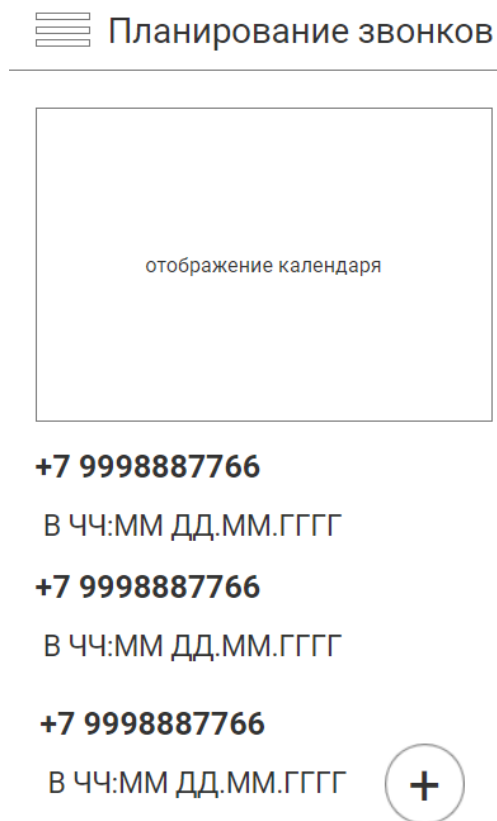


Рис. 6.6 – Планирование звонков

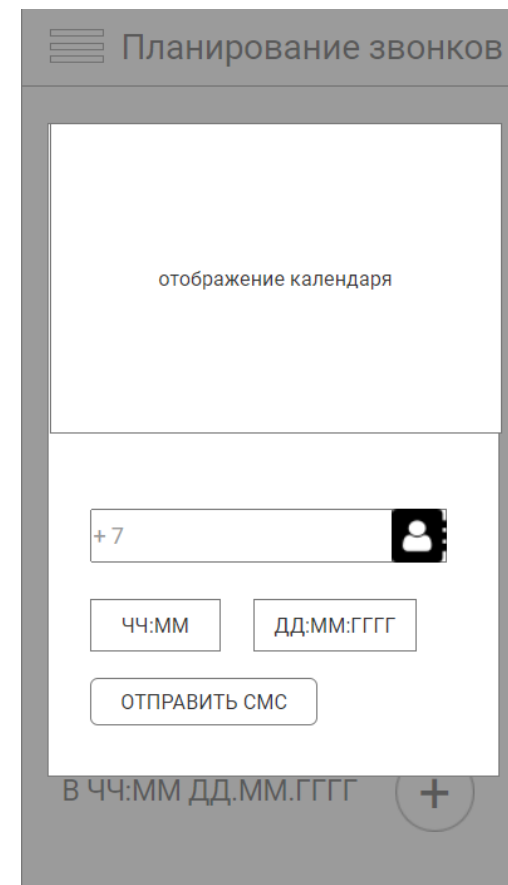


Рис. 6.7 – Добавление звонка

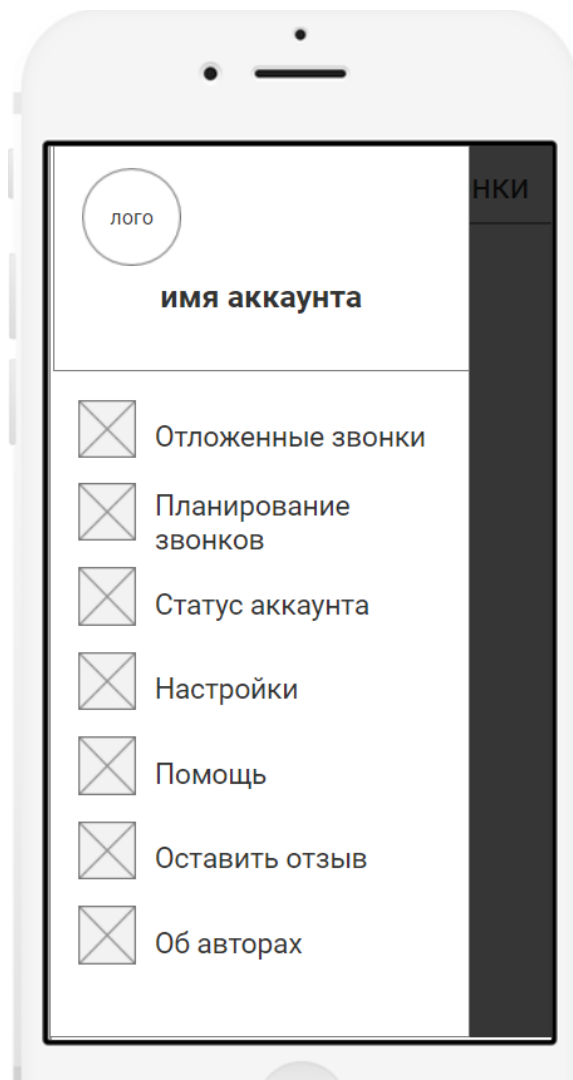


Рис. 6.8 – Боковое меню

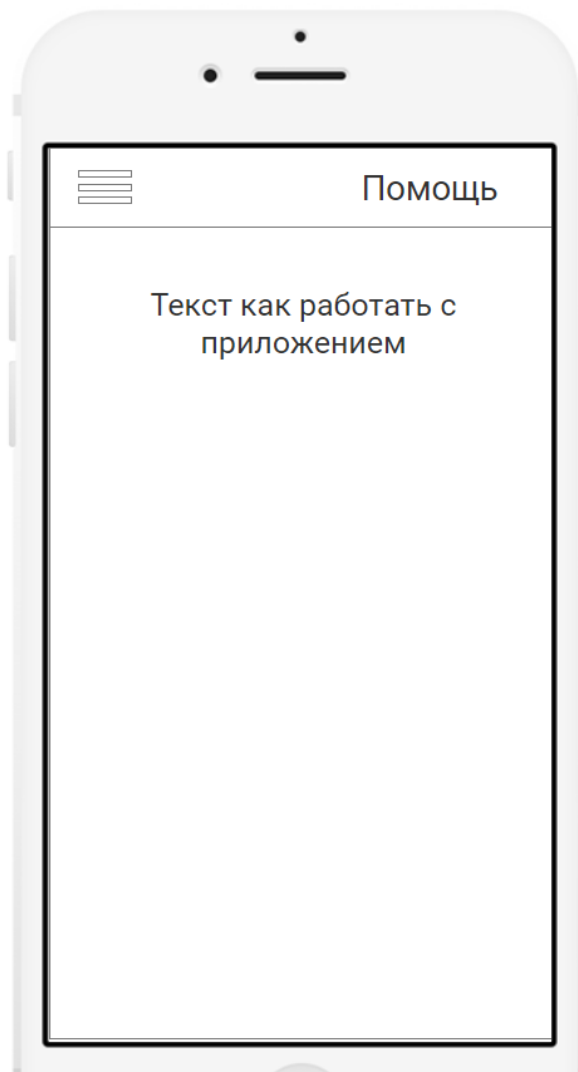


Рис. 6.9 – FAQ

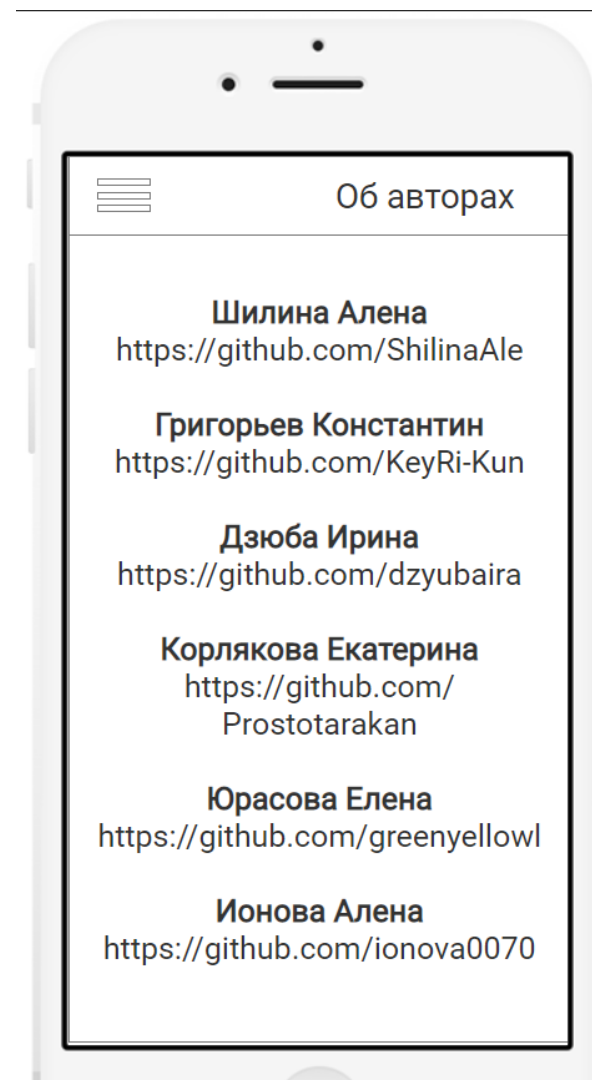


Рис. 6.10 – Об авторах

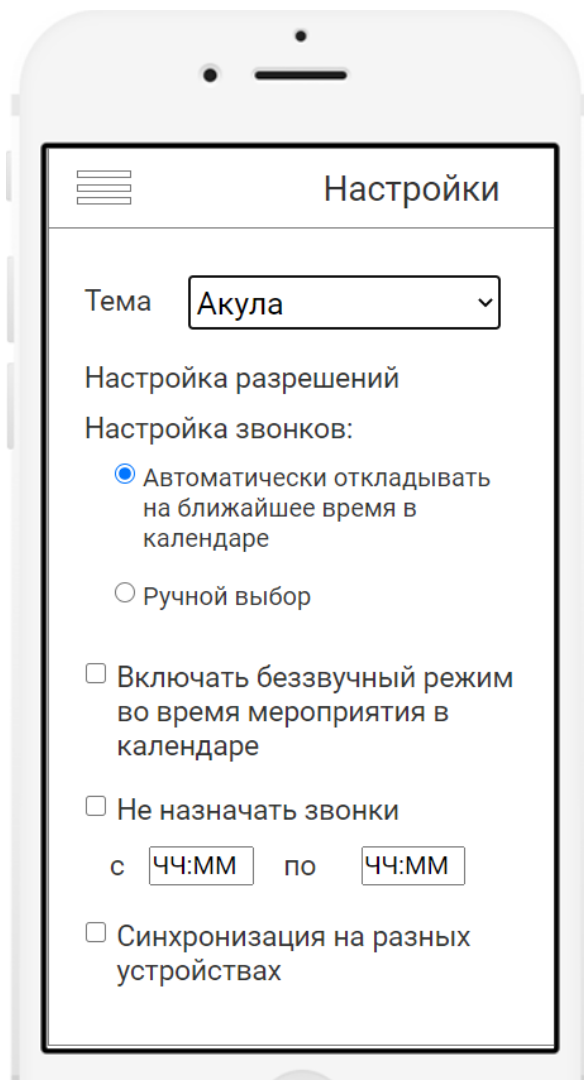


Рис. 6.11 – Настройки

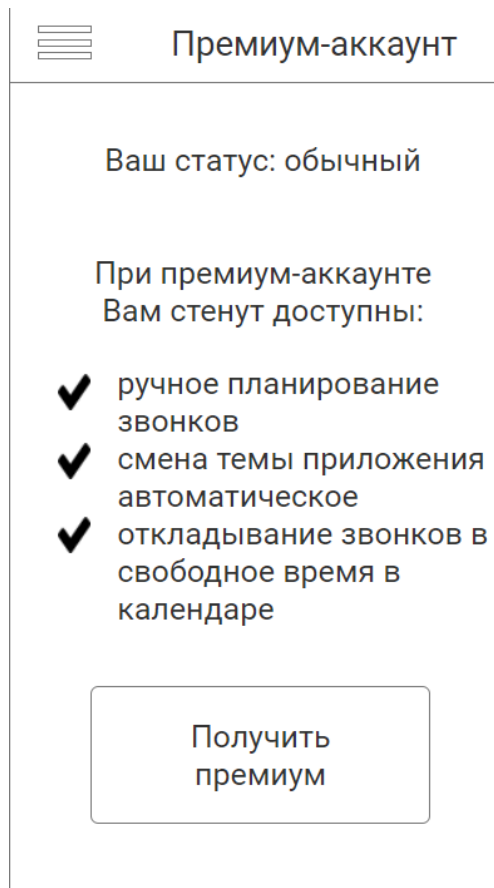


Рис. 6.12 – Статус

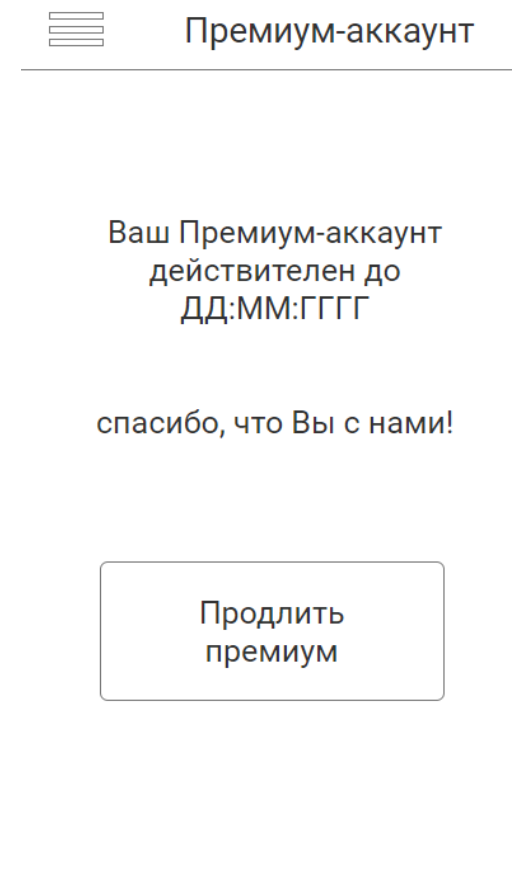


Рис. 6.13 – Получение премиума

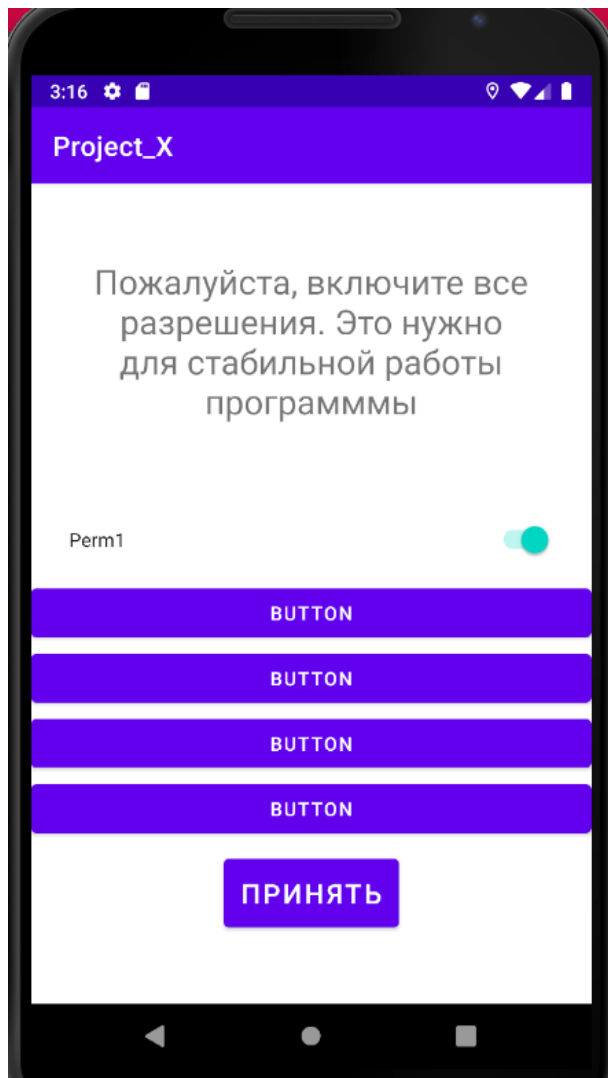


Рис. 6.14 – Разрешения

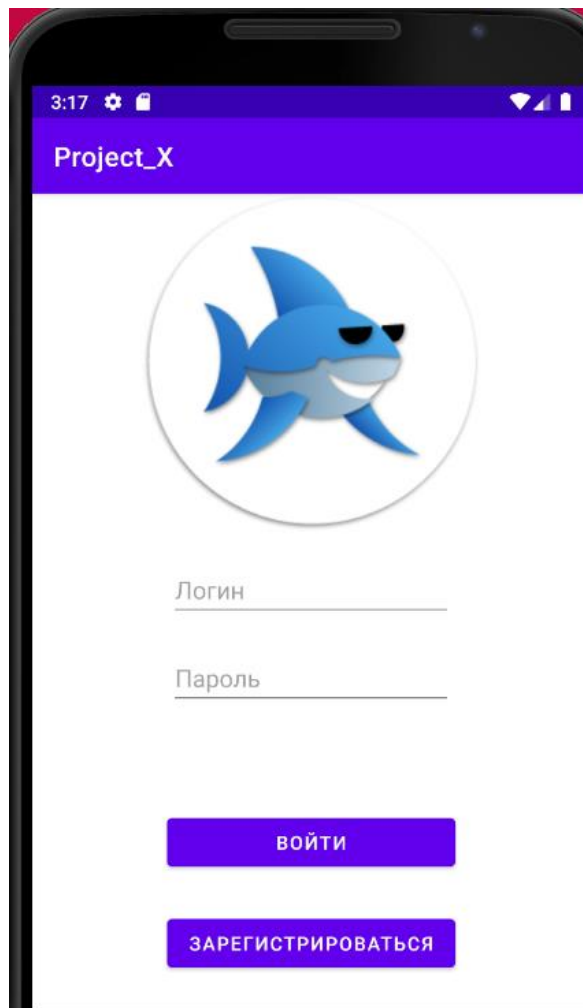


Рис. 6.15 – Авторизация

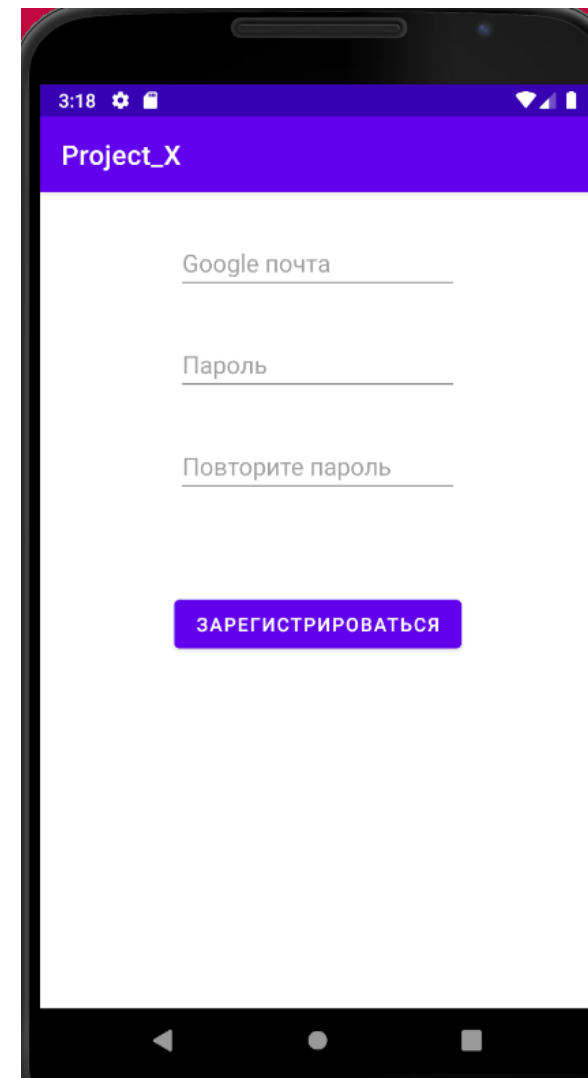


Рис. 6.16 – Регистрация

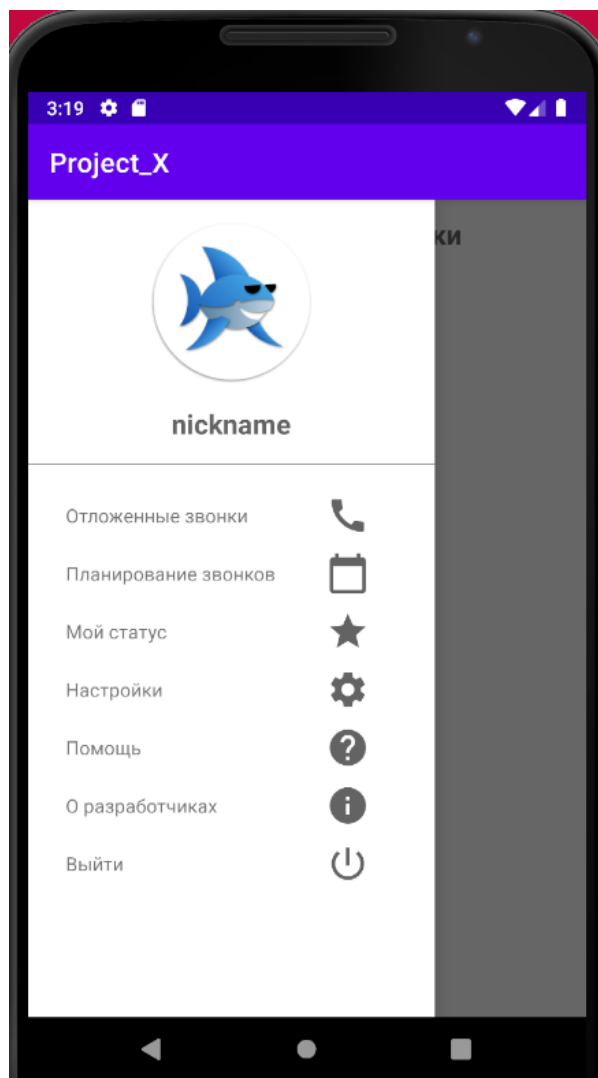


Рис. 6.17 – Боковое меню

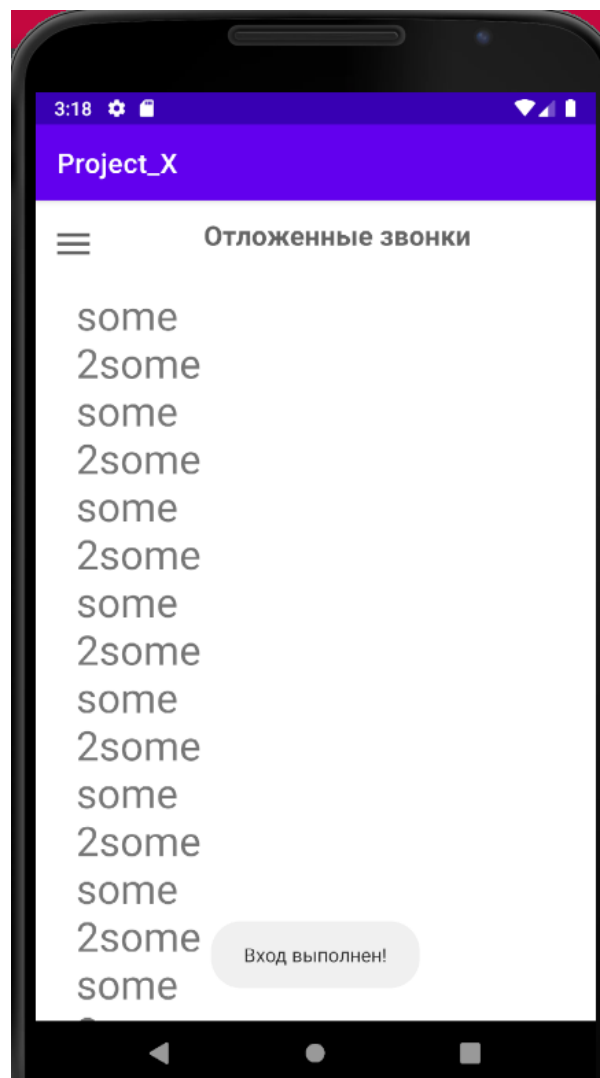


Рис. 6.18 – Планирование звонков

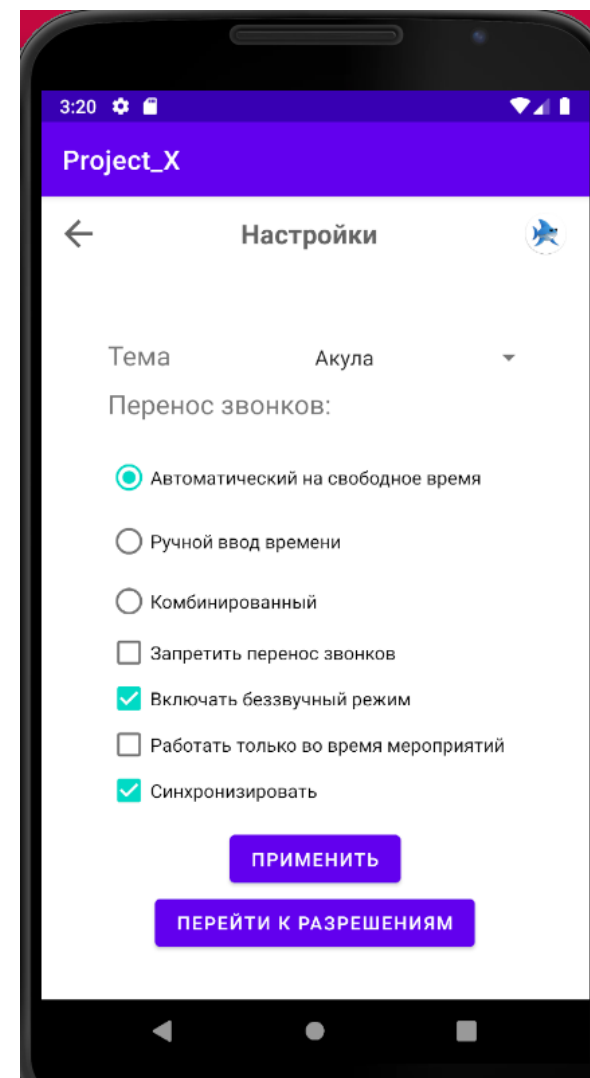


Рис. 6.19 – Настройки

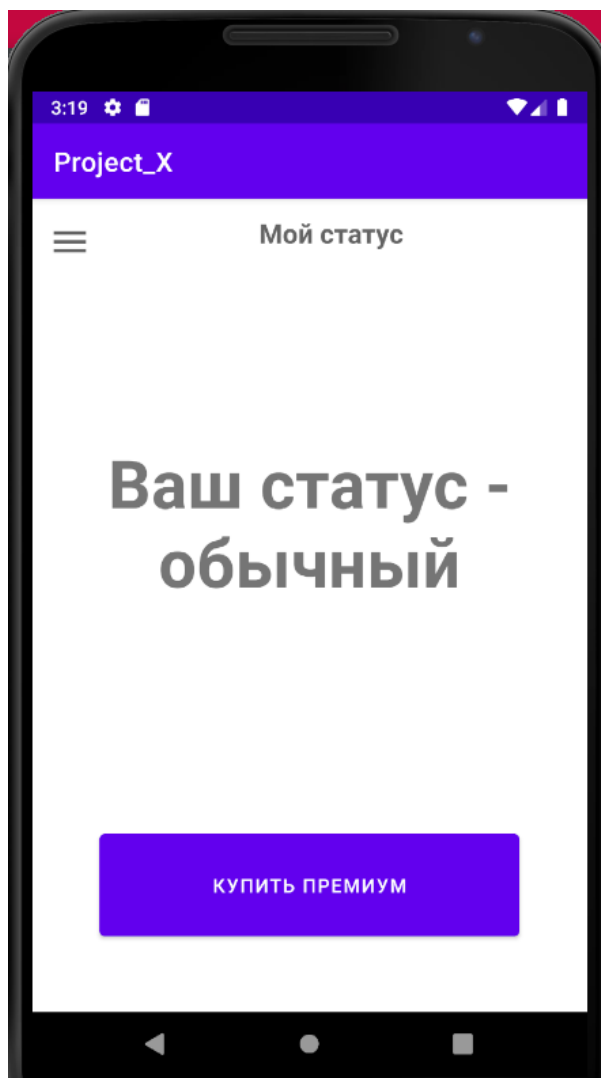


Рис. 6.20 – Статус

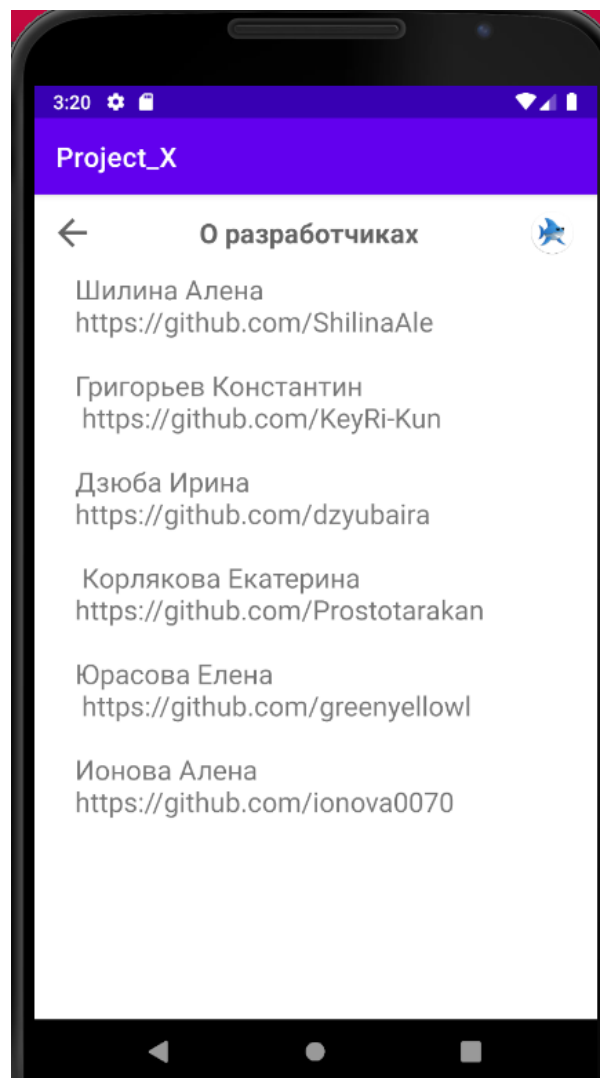


Рис. 6.21 – Об авторах

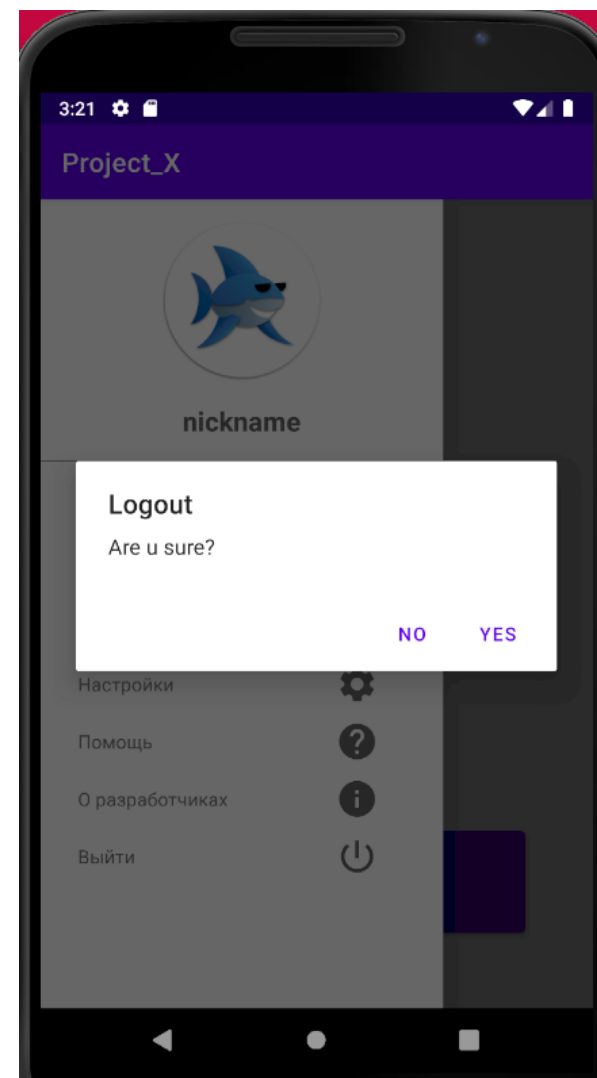


Рис. 6.22 – Выход

7. ОСНОВНЫЕ API КОМАНДЫ СЕРВЕРА

change_pass

Обработка смены пароля

Приходят данные:

email - почта пользователя

new_pass - новый пароль пользователя

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

delete_call

Обработка удаления звонка

Приходят данные:

id_call - ид звонка, который нужно удалить

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

get_rescheduling

Получение данных о перенесенных звонках пользователя с сейчас и дальше

Приходят данные:

email - почта пользователя

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

звонки: {phone: date_time}, где звонки - звонки, phone - номер на который нужно перезвонить, call_date_time - время звонка, callback_date_time - время, когда нужно перезвонить

login

Обработка авторизации

Приходят данные:

email - почта пользователя

пароль - пароль пользователя

android_id - идентификатор андроида

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

user_premium: 1 или 0 - есть премиум у пользователя или нет

set_premium

Обработка получения премиума

Приходят данные:

email - почта пользователя

summ - сумма платежа

pay_date - дата платежа в формате дд.мм.гггг

pay_time - время платежа в формате чч: мм

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

set_rescheduling

Обработка переноса звонка

Приходят данные:

email - почта пользователя

recipient_number - номер входящего звонка

call_date - дата звонка в формате дд.мм.гггг

call_time - время звонка в формате чч: мм

callback_date - дата, на которую переносится звонок в формате дд.мм.гггг

callback_time - время, которое переносится звонок в формате чч: мм

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

set_setting

Обработка изменения настройки устройства

Приходят данные:

email - почта пользователя

имя_устройства - идентификатор андроида

setting_name - название настройки. Могут быть: color_logo, rescheduling_in_event, rescheduling_out_event.

value - значение настройки, которое необходимо установить

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

sign_up

Обработка регистрации

Создаются новые записи в таблицах пользователей, номеров, user_device и устройств. В последнюю таблицу записываются стандартные настройки

Приходят данные:

email - почта пользователя

phone - номер телефона

пароль - пароль пользователя

android_id - идентификатор андроида. Уникален и создается на стороне клиента.

Возвращает JSON с полями:

ошибка: 1 или 0

error_text: пустая строка или текст ошибки

8. МОДЕЛЬ БАЗЫ ДАННЫХ

База данных приложения (БД) включает в себя 6 таблиц: пользователи, номера телефонов, устройства, звонки, платежи и таблица связей пользователей и устройств, таблицы 8.1-8.6. На рисунке 8.1 общая схема базы данных и соотношения таблиц.

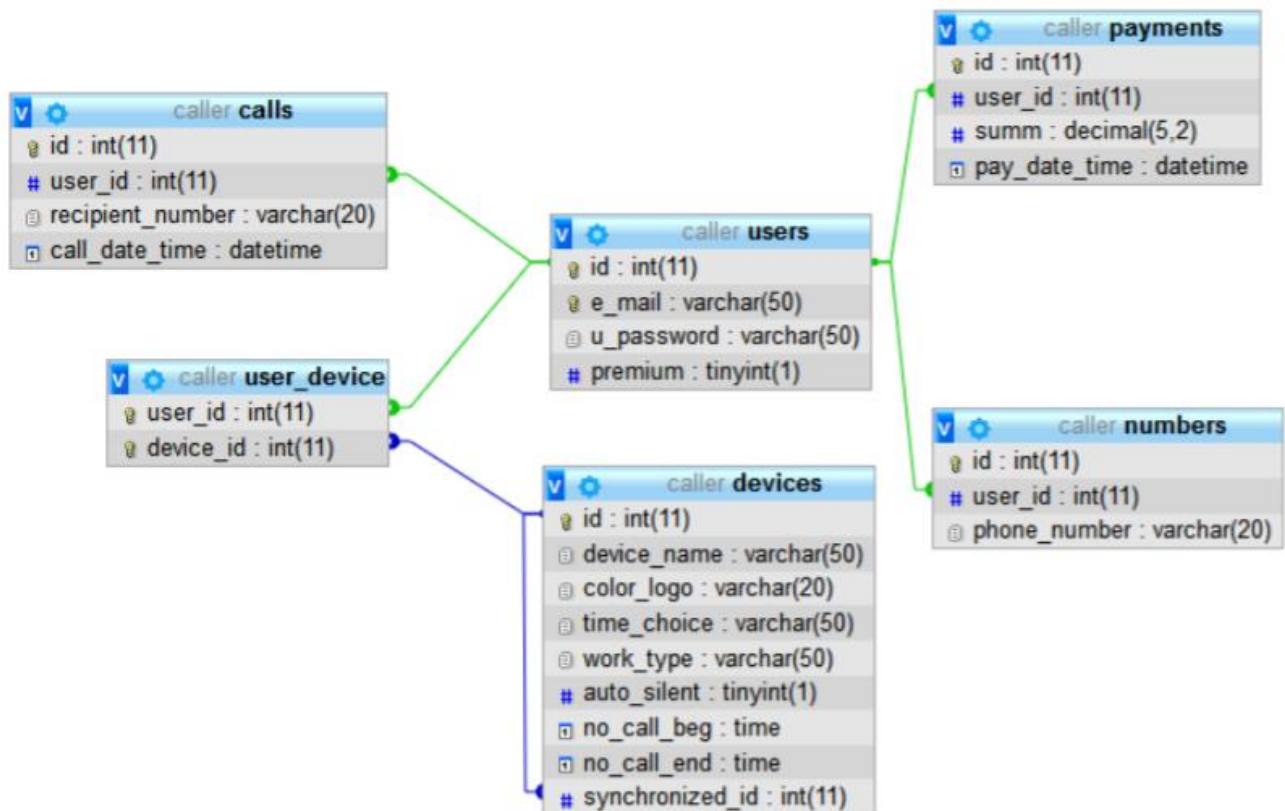


Рис. 8.1 – Схема БД и связи отдельных таблиц

Для каждой таблицы определены атрибуты, их описание, тип значения, ключи и примеры значений.

Необходимость использовать таблицу для связи пользователей и мобильных устройств, а также разделение телефонных номеров и мобильных устройств обуславливается тем, что один и тот же пользователь может установить приложение на несколько своих устройств или поменять сим-карту, тогда как данные об оплате премиум-аккаунта или о переносе звонков должны сохраняться независимо от мобильного устройства или номера телефона.

Таблица 8.1 – Пользователи (Users)

Атрибут	Описание	Тип	Ключ	Примеры значений
id	Идентификатор пользователя	int	PK	«1», «2» ...
e_mail	Электронный адрес	varchar (50)		«ivanivanov@gmail.com», ...
u_password	Пароль	varchar (50)		«Ff(54-.Kk», ...
premium	Наличие премиум-аккаунта	bool		«0» (нет премиум-аккаунта), «1» (есть премиум-аккаунт)

Таблица 8.2 – Номера телефонов (Numbers)

Атрибут	Описание	Тип	Ключ	Примеры значений
id	Идентификатор номера телефона	int	PK	«1», «2» ...
user_id	Идентификатор пользователя	int	FK (Users: id)	«1», «2» ...
phone_number	Номер телефона	varchar (20)		«89001234567», «89859876543», ...

Таблица 8.3 – Устройства (Devices)

Атрибут	Описание	Тип	Ключ	Примеры значений
id	Идентификатор устройства	int	PK	«1», «2» ...
device_name	Название устройства	nvarchar (50)		«Samsung Galaxy A12», «Xiaomi Redmi Note 8T», «Телефон 1», ...
color_logo	Цветовая схема и логотип	nvarchar (20)		«Shark», «Raccoon», «Panther»
rescheduling_in_event	Вариант переноса во время мероприятия	nvarchar (50)		«Auto», «Hand»
rescheduling_out_event	Вариант переноса вне мероприятия	nvarchar (50)		«None», «Hand»

Таблица 8.4 – Номер – устройство (UserDevice)

Атрибут	Описание	Тип	Ключ	Примеры значений
user_id	Идентификатор пользователя	int	FK (Users: id)	«1», «2» ...
device_id	Идентификатор устройства	int	FK (Devices: id)	«1», «2» ...

Таблица 8.5 – Звонки (Calls)

Атрибут	Описание	Тип	Ключ	Примеры значений
id	Идентификатор звонка	int	PK	«1», «2», ...
user_id	Идентификатор пользователя, запланировавшего звонок	int	FK (Users: id)	«1», «2» ...
recipient_number	Номер телефона, по которому запланирован звонок	varchar (20)		«89001234567», «89859876543», ...
call_date_time	Дата и время поступившего пользователю звонка в формате UNIX	varchar (20)		«1574424627», ...
callback_date_time	Дата и время, которые пользователь выбирает при переносе, чтобы перезвонить в формате UNIX	varchar (20)		«1574424627», ...

Таблица 8.6 – Пользователи (Users)

Атрибут	Описание	Тип	Ключ	Примеры значений
id_payment	Идентификатор платежа	int	PK	«1», «2», ...
user_id	Идентификатор пользователя, запланировавшего звонок	int	FK (Users: id)	«1», «2» ...
summ	Сумма платежа	decimal (5, 2)		«100,00», ...
pay_date_time	Дата и время платежа	datetime		«01.05.2021 14:00», ...

9. РЕЗУЛЬТАТЫ РАЗРАБОТКИ ПРОЕКТА

Использование проектной документации при разработке проекта позволило создать мобильное приложение, отвечающее заданным требованиям и имеющее оговоренный функционал. В приложениях А-В листинг программного кода frontend, backend, сервера. В рамках выполнения проекта сервер использован локальный, что позволяет протестировать все функции, но требует развертывания сервера на компьютере и выставления его параметров в классе ServerHandler приложения.

9.1 Архитектура приложения

Одним из элементов описания и структурирования проекта является его архитектура.

Для мобильного приложения упрощенное представление об архитектуре системы включает только три элемента – клиентская часть, серверная часть и базу данных, однако, при дальнейшей декомпозиции и усложнении структуры можно выделить важные элементы разрабатываемого приложения и их связи. На рисунках 9.1.1 и 9.1.2 системная архитектура проекта, на рисунке 9.1.3 – участники процесса эксплуатации проекта.

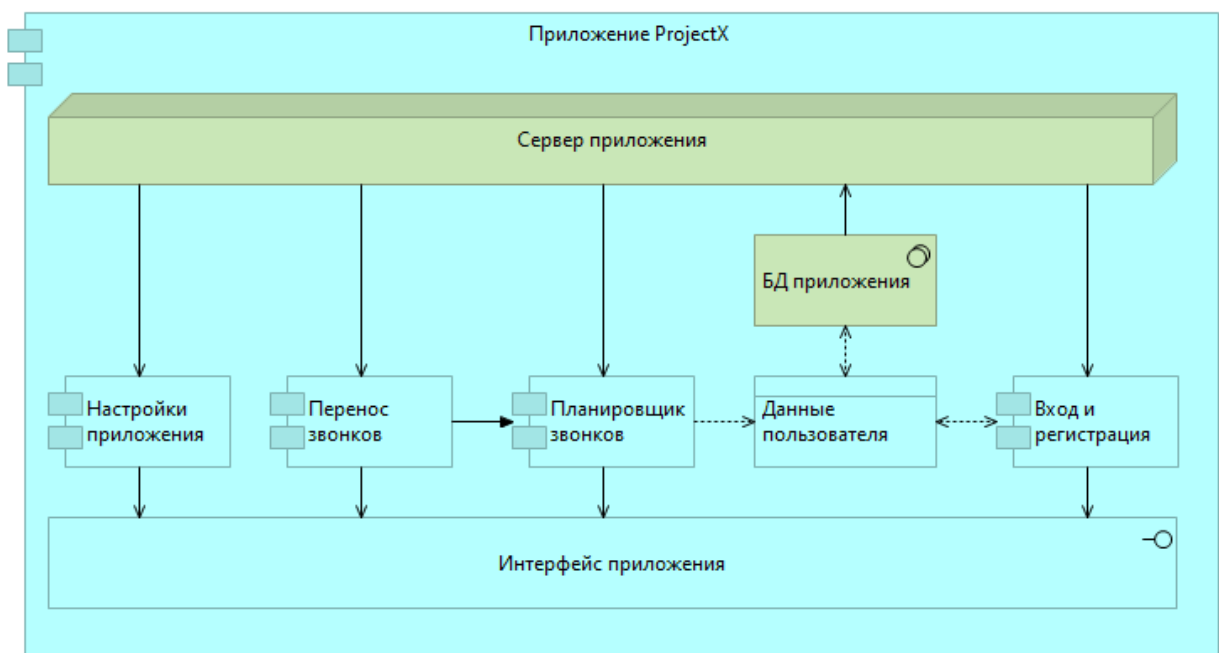


Рис. 9.1.1 – Архитектура приложения

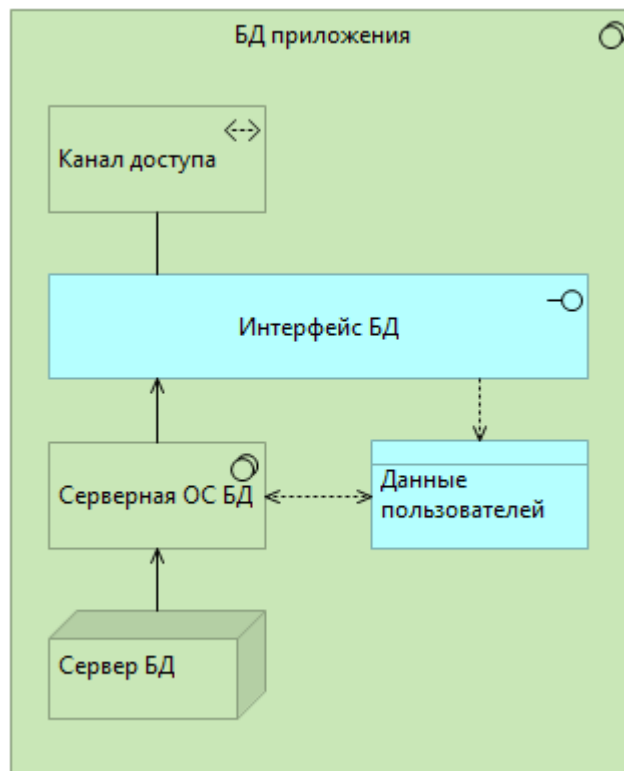


Рис. 9.1.2 – Архитектура БД приложения

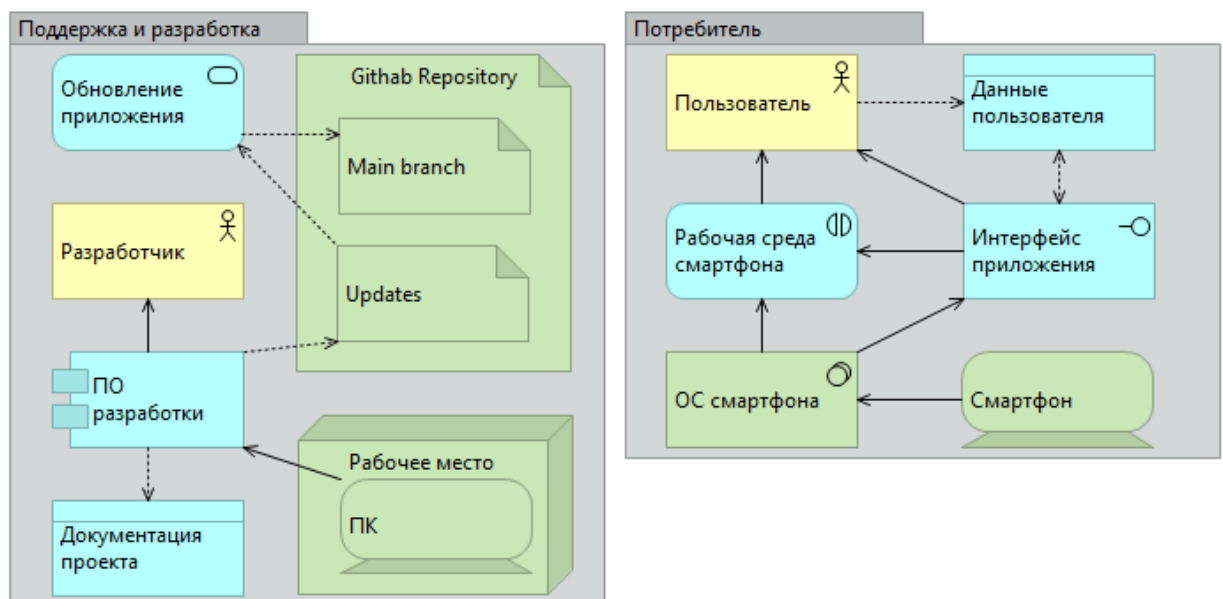


Рис. 9.1.3 – Взаимодействие приложения и пользователя или разработчика

9.2. Клиентский интерфейс

На основе прототипа был доработан дизайн мобильного приложения. При этом схема его структуры осталась такой, как изображена на рисунке 6.1, где переключение между основными экранными формами осуществляется с помощью навигационного меню. В рамках прототипирования приложения были разработаны макеты всех окон кроме

окна оплаты при покупке премиума, рисунок 9.2.1, необходимость в которой возникла при разработке модуля оплаты.

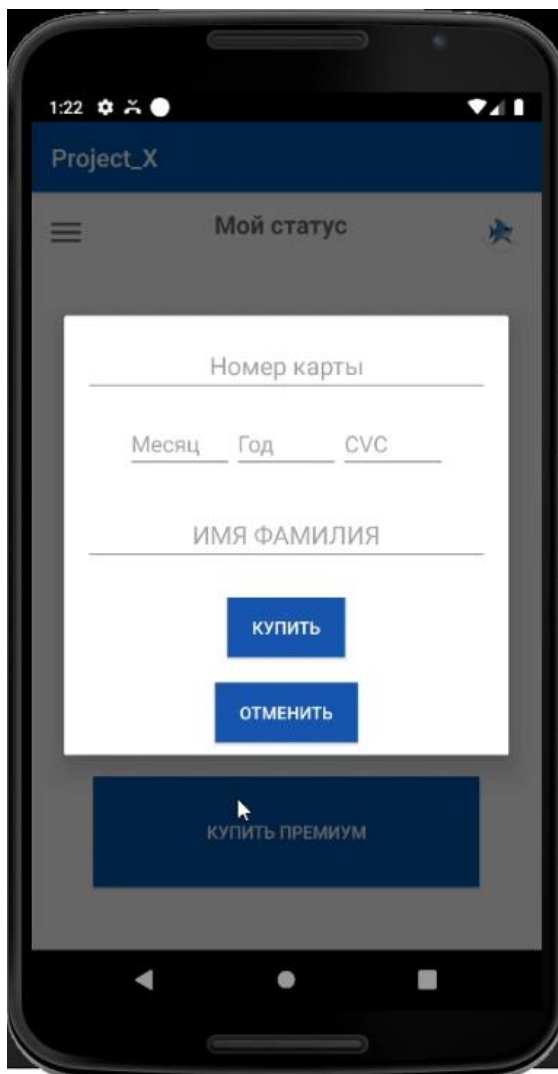


Рис. 9.2.1 – Окно оплаты премиум-подписки

С клиентской стороны также реализованы проверки на вводимые данные. При регистрации, авторизации и покупке премиум-аккаунта. При вводе логина не gmail, при несовпадении паролей, а также невозможных полей «месяц» и «год» появляется сообщение о том, что необходимо проверить введенные данные. На сервер отправляется только корректная информация.

Специально для мобильного приложения была разработана его цветовая гамма. вокруг нас каждый день мелькает множество оттенков и полутонов. Одни вызывают плохие эмоции или агрессию, другие поднимают настроение, успокаивают. Синий — это основной цвет из всех холодных

цветов спектра. Использование холодных цветов в дизайне помогает создать ощущение спокойствия. Так, основная тема приложения была выбрана в синих оттенках — «Акула», рисунок 9.2.2.

```
<color name="sharkPrimary">#1455af</color>  
<color name="sharkHead">#0e4187</color>  
<color name="sharkGrey">#666666</color>  
<color name="sharkWhite">#FFFFFF</color>
```

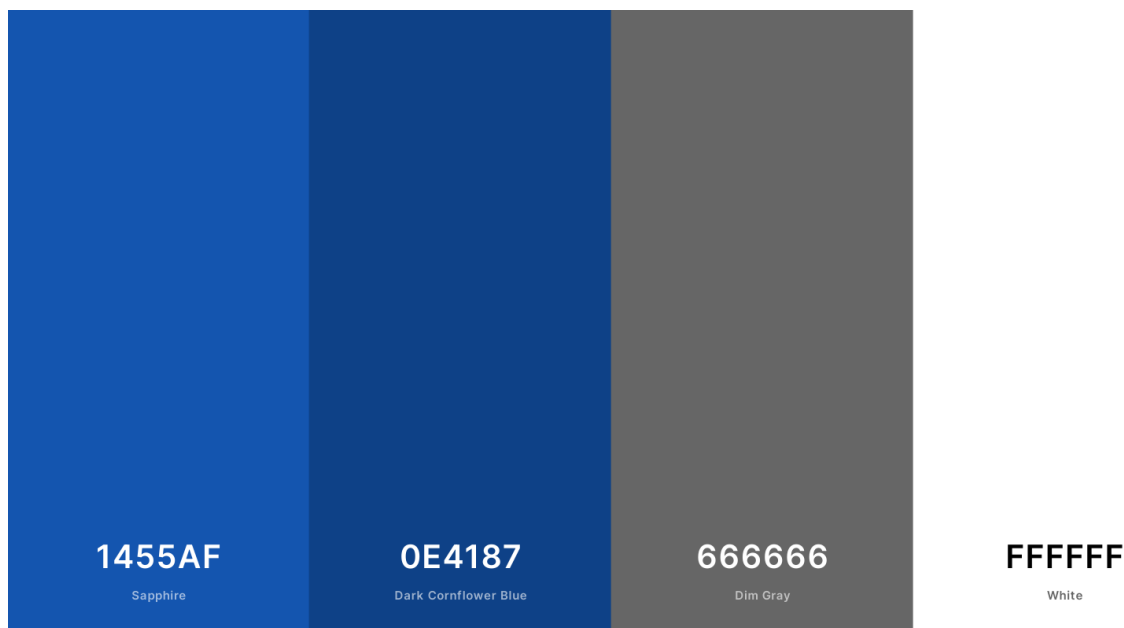


Рис. 9.2.2 – Цветовая схема «Акула»

Такой дизайн является базовым для приложения, и при установке приложения используется именно он.

На рисунке 9.2.3 пример работы приложения с цветовой схемой «Акула».

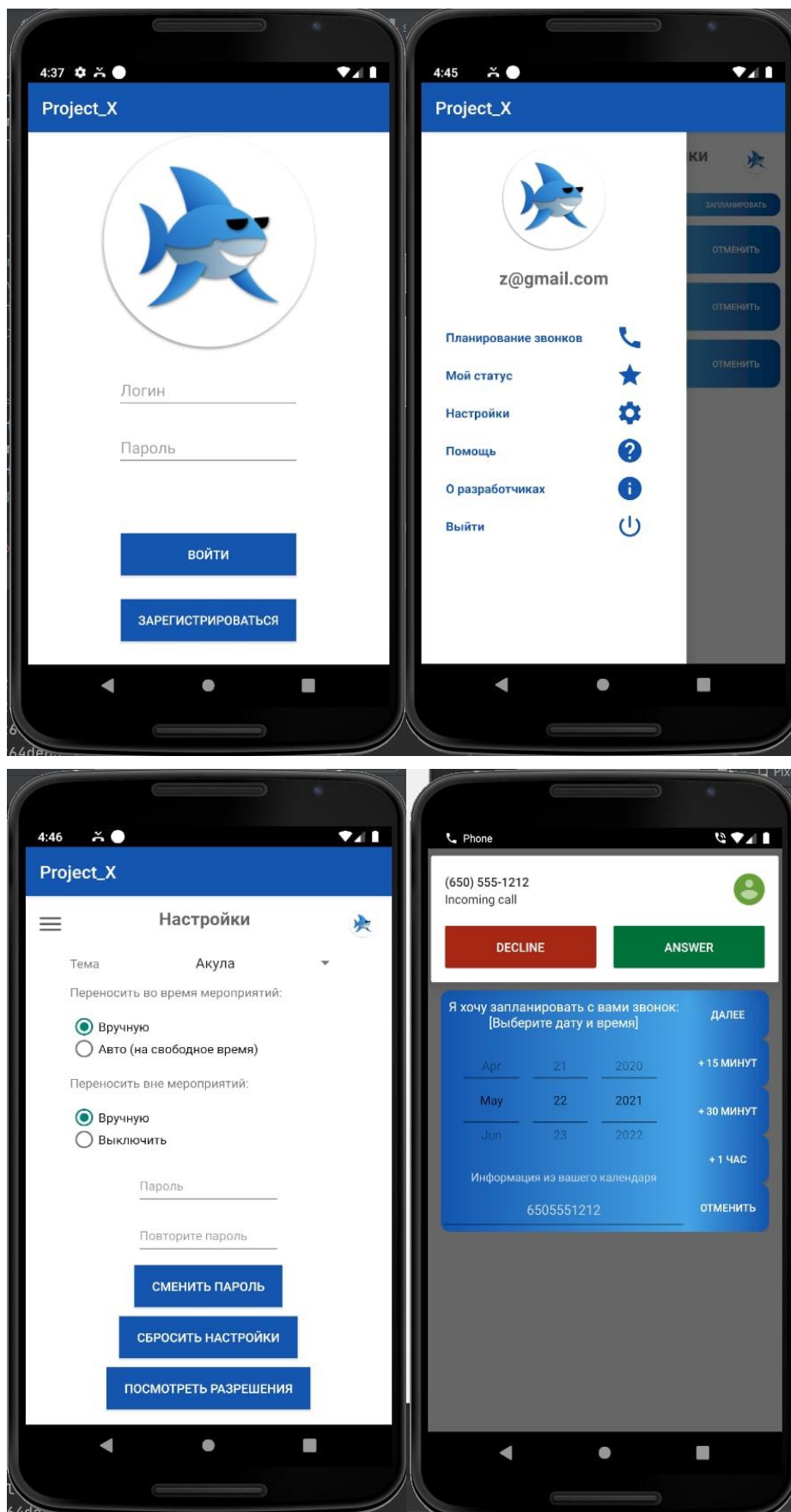


Рис. 9.2.3 – Приложение с цветовой схемой «Акула»

Для премиум-пользователей доступны еще две темы - «Пантера» и «Енот».

Для быстрой смены темы по нажатию кнопки была реализована ссылочная система переключения. Каждый элемент в приложении ссылается на атрибут цвета, например:

```
android:textColor="?attr/colorText"/>
```

или

```
android:background="?attr/colorDrawerBody"
```

Атрибуты находятся в файле styles.xml:

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme - shark -->
    <style name="Theme.Project_X" parent="Theme.AppCompat.DayNight.DarkActionBar">

        <item name="colorPrimary">@color/sharkPrimary</item>
        <item name="colorPrimaryVariant">@color/sharkHead</item>
        <item name="colorText">@color/sharkGrey</item>
        <item name="colorButton">@color/sharkPrimary</item>
        <item name="colorTextOnButton">@color/sharkWhite</item>
        <item name="colorBackground">@color/sharkWhite</item>
        <item name="colorOnPrimary">@color/sharkWhite</item>
        <item name="colorDrawerTextHead">@color/sharkGrey</item>
        <item name="colorDrawerBody">@color/white</item>
        <item name="colorDrawerTextBody">@color/sharkPrimary</item>
```

Этот файл задает основные элементы приложения, такие как фон приложения, текст на кнопках или градиент, используемый при планировании звонка. И уже эти элементы описаны в каждой из тем в файле themes.xml:

```

<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme - shark -->
    <style name="Theme.Project_X" parent="Theme.AppCompat.DayNight.DarkActionBar">

        <item name="colorPrimary">@color/sharkPrimary</item>
        <item name="colorPrimaryVariant">@color/sharkHead</item>
        <item name="colorText">@color/sharkGrey</item>
        <item name="colorButton">@color/sharkPrimary</item>
        <item name="colorTextOnButton">@color/sharkWhite</item>
        <item name="colorBackground">@color/sharkWhite</item>
        <item name="colorOnPrimary">@color/sharkWhite</item>
        <item name="colorDrawerTextHead">@color/sharkGrey</item>
        <item name="colorDrawerBody">@color/white</item>
        <item name="colorDrawerTextBody">@color/sharkPrimary</item>
    
```

Таким образом, для использования трех различных тем, необходимо прописать только файл themes.xml, изменив только ссылки на цвета:

```

<!-- Base application theme. -->
<style name="Theme.Project_X_pantera" parent="Theme.AppCompat.DayNight.DarkActionBar">

    <item name="colorPrimary">@color/panteraPrimary</item>
    <item name="colorPrimaryVariant">@color/panteraHead</item>
    <item name="colorText">@color/panteraDrawer</item>
    <item name="colorButton">@color/panteraPrimary</item>
    <item name="colorTextOnButton">@color/panteraWhite</item>
    <item name="colorBackground">@color/panteraBack</item>
    <item name="colorOnPrimary">@color/panteraWhite</item>
    <item name="colorDrawerTextHead">@color/panteraWhite</item>
    <item name="colorDrawerBody">@color/panteraDrawer</item>
    <item name="colorDrawerTextBody">@color/panteraPrimary</item>

```

При смене цветовой гаммы приложения меняется также и иконка, используемая на экранных формах и с появлением иконки во время звонка.

Темная тема приложения представлена темой «Пантера», палитра на рисунке 9.2.4.

Третья тема, «Енот», выполнена в кофейных тонах, с добавлением серого и белого для отображения текста и мелких деталей интерфейса, рисунок 9.2.5. Примеры приложения с разными темами на рисунках 9.2.6-9.2.7.

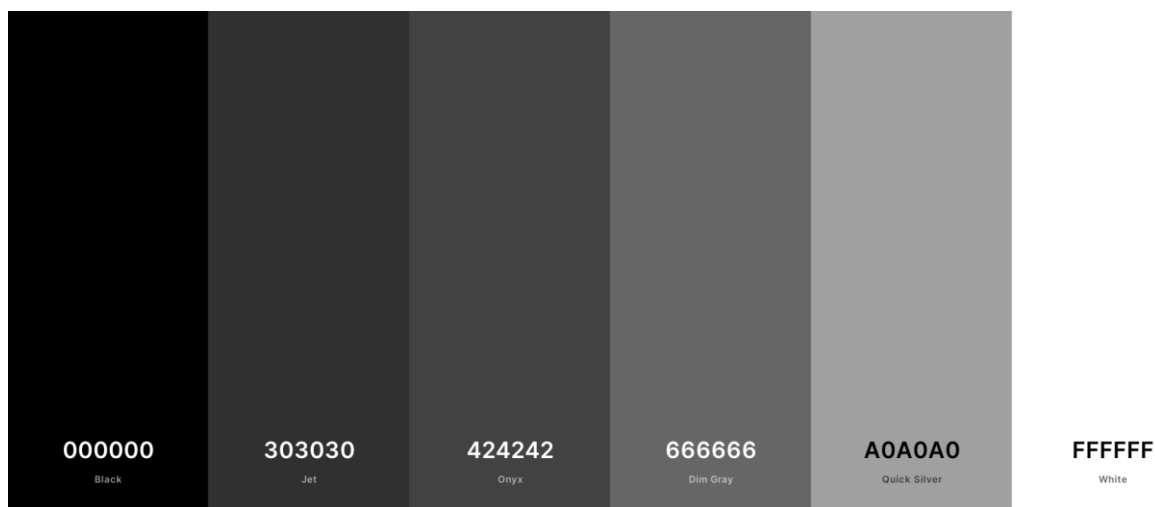


Рис. 9.2.4 – Приложение с цветовой схемой «Пантера»

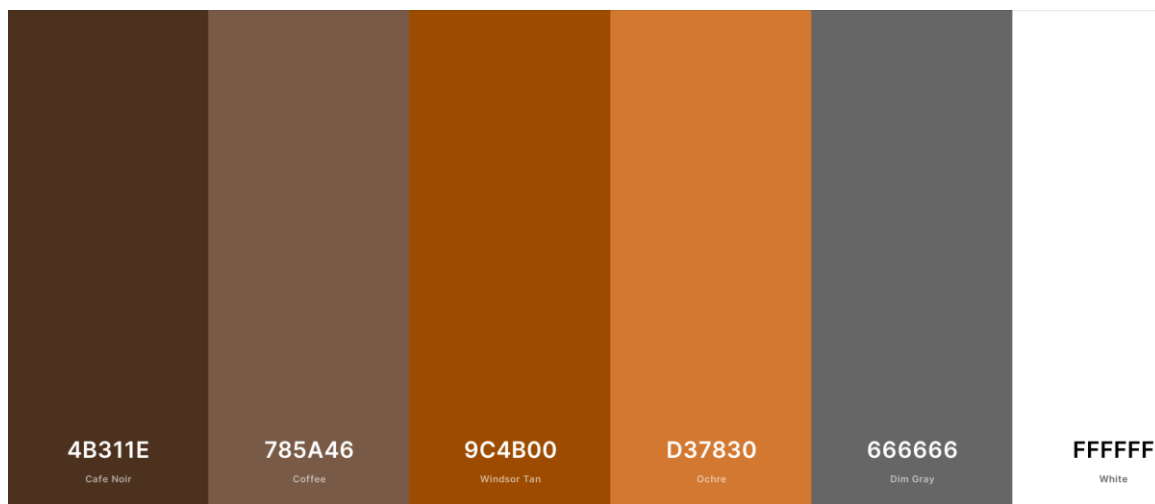


Рис. 9.2.5 – Приложение с цветовой схемой «Енот»

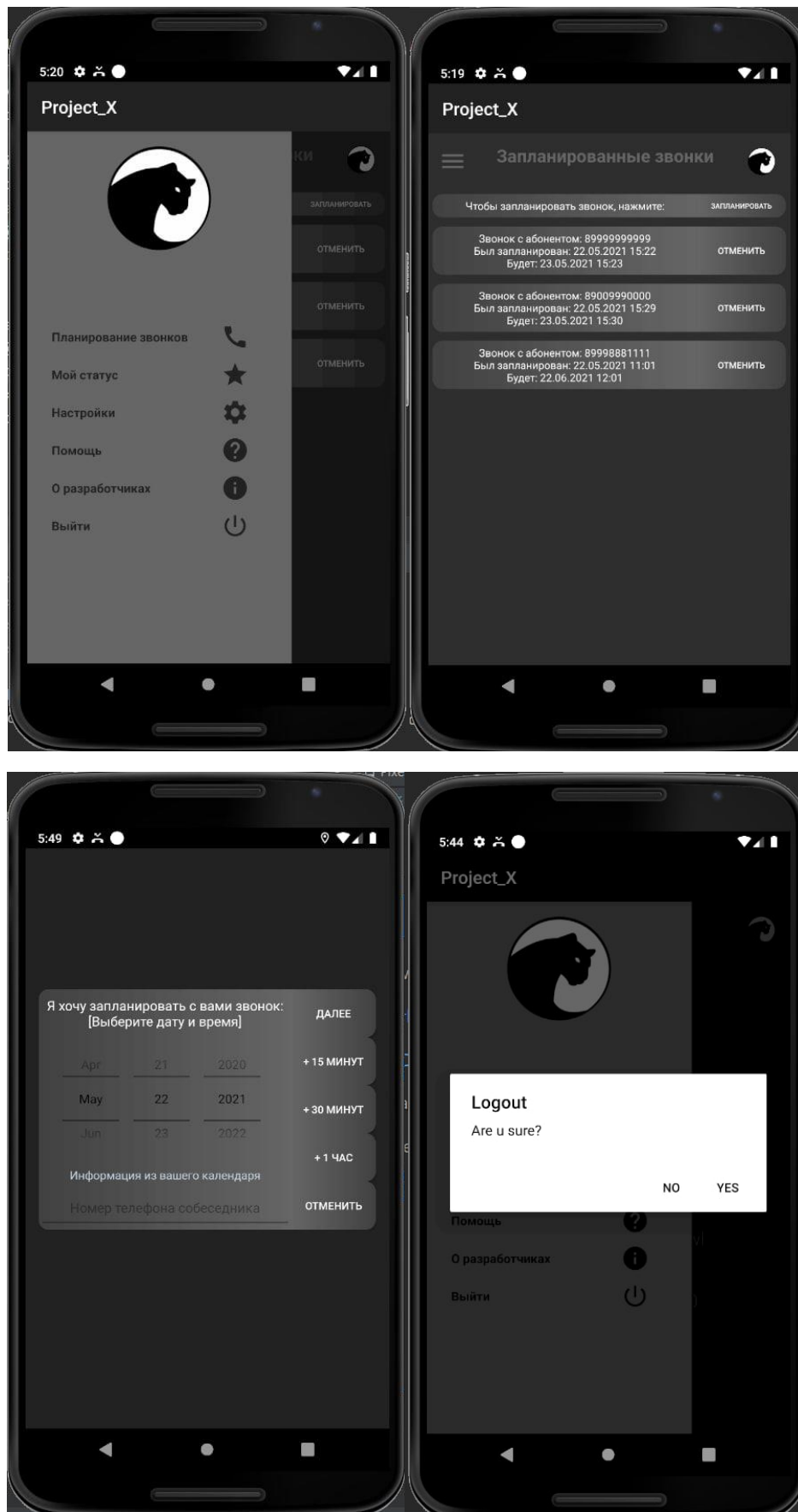


Рис. 9.2.6 – Приложение с цветовой схемой «Пантера»

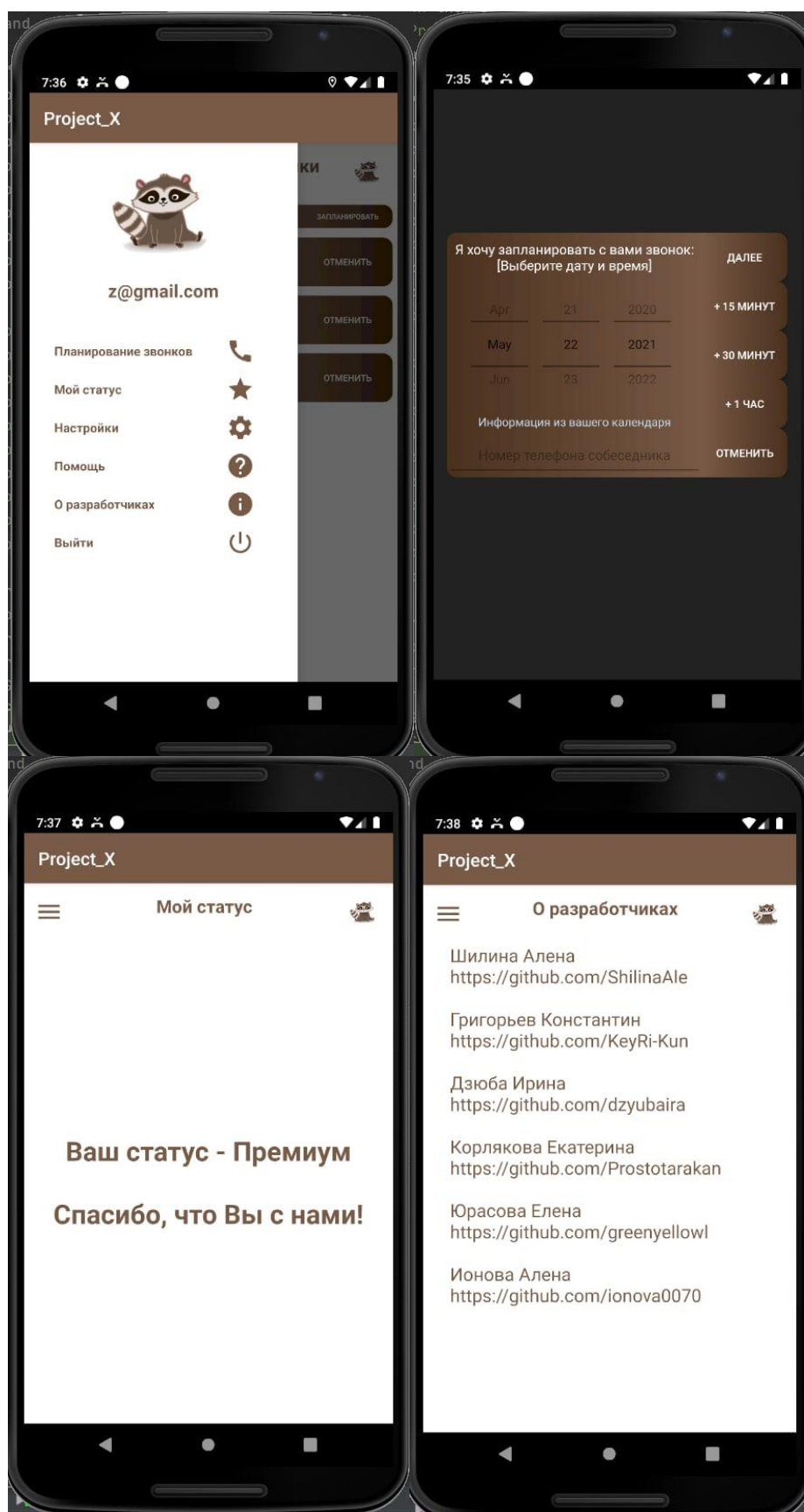


Рис. 9.2.7 – Приложение с цветовой схемой «Енот»

9.3 Тестирование приложения

В рамках проведения ручного тестирования с использованием разработанных тесткейсов (раздел 5) было проведено 18 тестирований в ручном режиме всех функций мобильного приложения. Получены следующие результаты:

Количество тестов: 18

Успешные тесты: 18

Неудачные тесты: 0

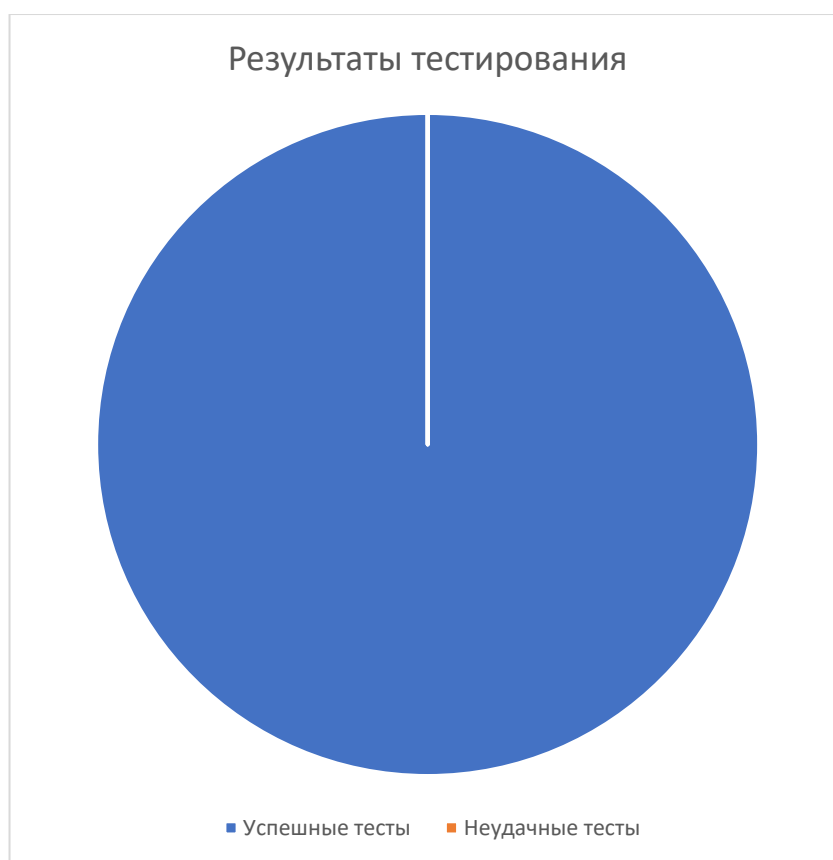
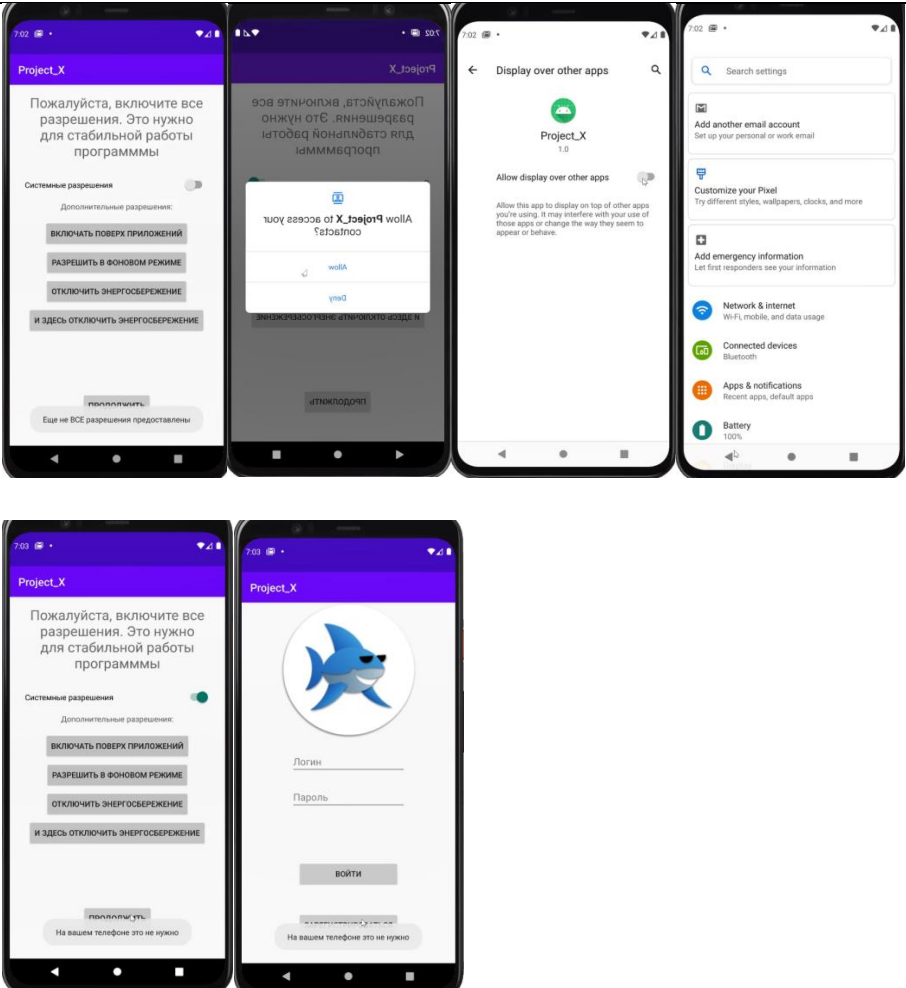


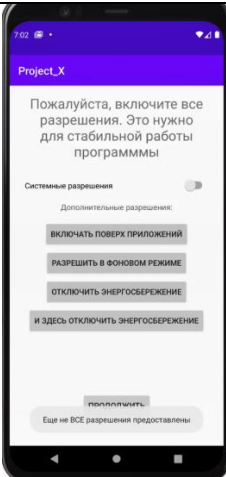
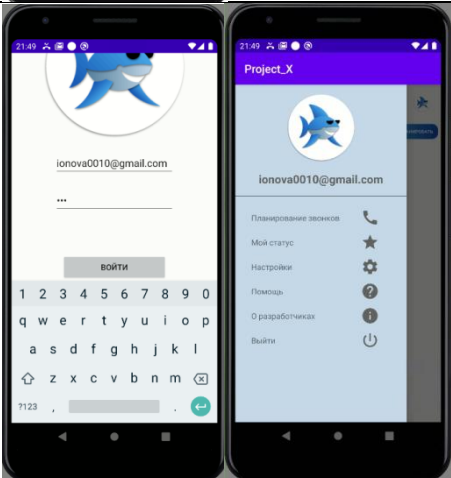
Рис. 9.3.1 – Результаты тестирования.



В таблице 9.3.1 для каждого из проведенных тестов приведены скриншоты тестирования и его результаты.

.

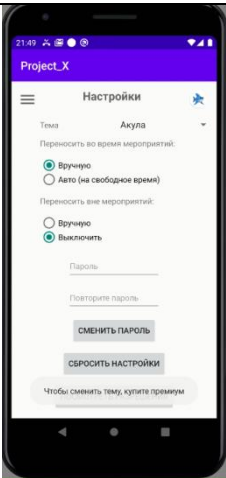
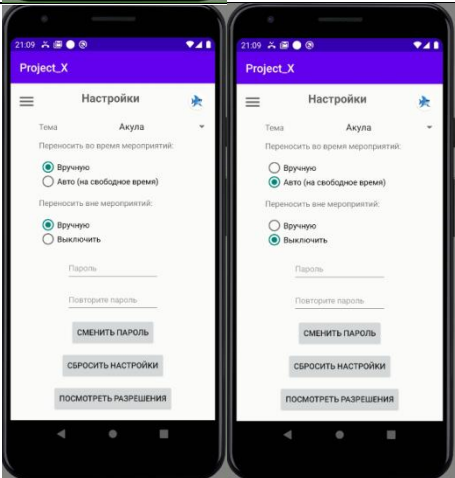
Таблица 9.3.1 – Результаты тестирования

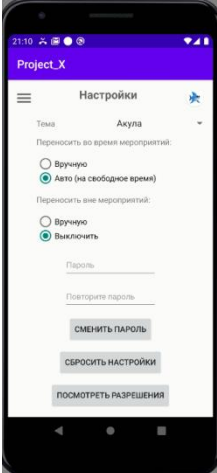
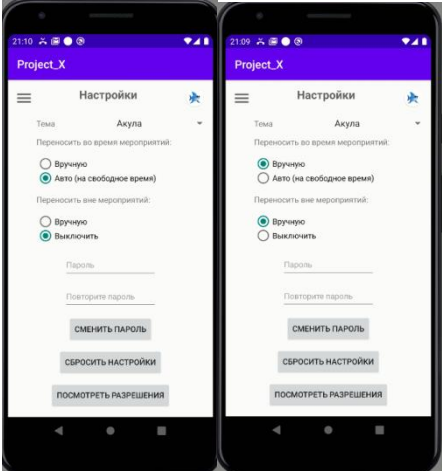
ID теста	Название теста	Скриншоты	Результат
001	Принятие разрешений после установки		Пройден

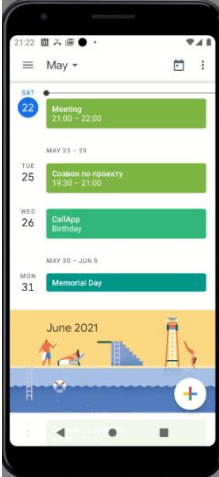
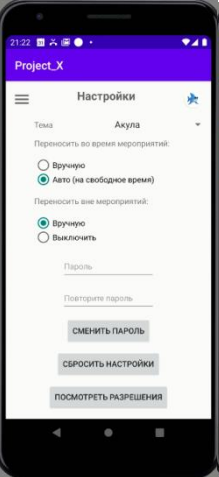

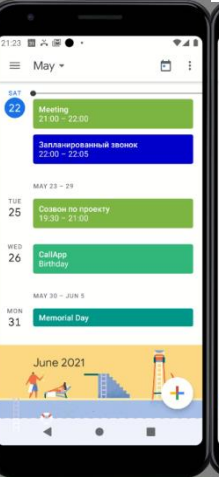
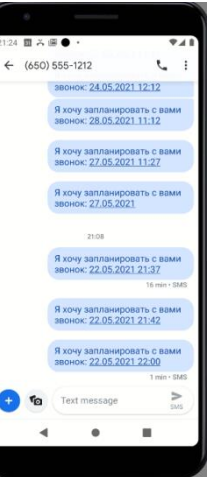
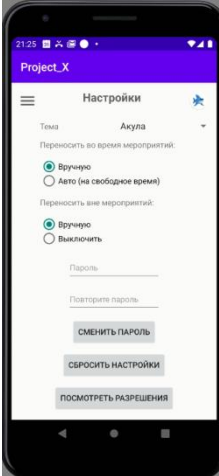
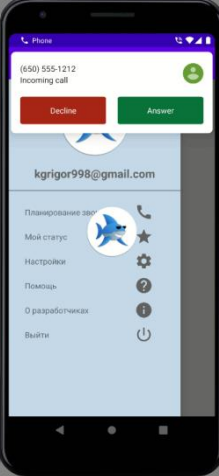


ID теста	Название теста	Скриншоты	Результат
002	Принятие не всех разрешений		Пройден
101	Успешная авторизация пользователя		Пройден

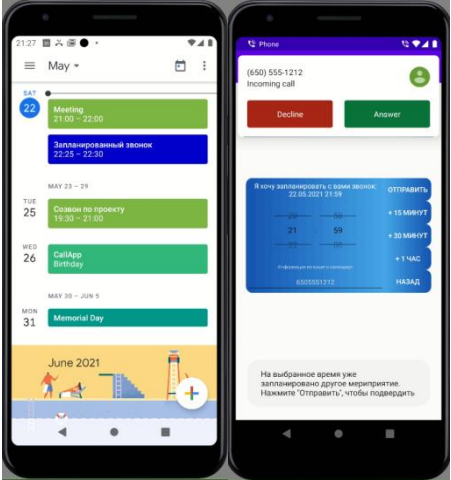
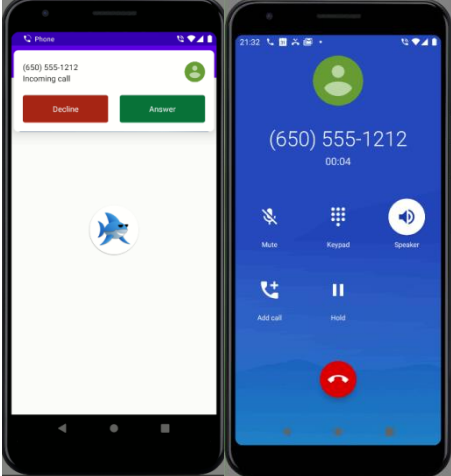
ID теста	Название теста	Скриншоты	Результат
102	Неудачная авторизация пользователя (несуществующий email)		Пройден
103	Неудачная авторизация пользователя (неверный пароль)		Пройден

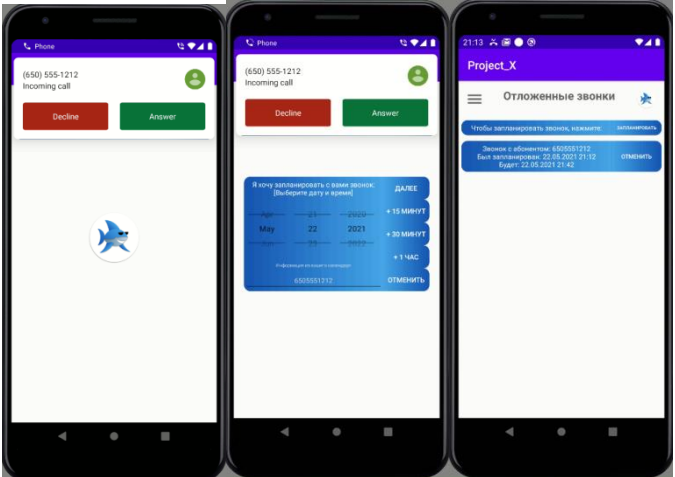
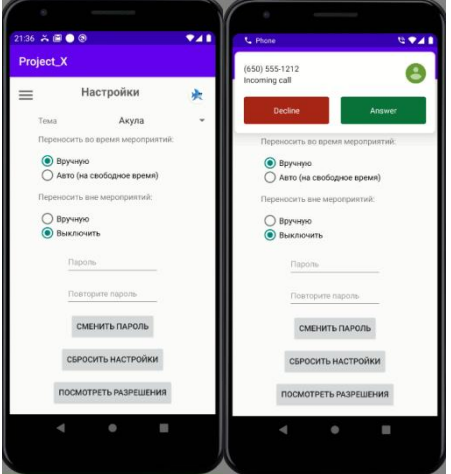
ID теста	Название теста	Скриншоты	Результат
201	Покупка премиум-аккаунта		Пройден
301	Успешная смена цветовой схемы		Пройден

ID теста	Название теста	Скриншоты	Результат
302	Неудачная смена цветовой схемы		Пройден
4501	Настройка работы алгоритма и варианта использования	 <p>После выхода из настроек и входа:</p>	Пройден

ID теста	Название теста	Скриншоты	Результат
			
4502	Сброс настроек		Пройден

ID теста	Название теста	Скриншоты	Результат
701	Перенос звонка во время мероприятий (автоматический)	    	Пройден
702	Успешный перенос звонка во время мероприятий (ручной)	   	Пройден

ID теста	Название теста	Скриншоты	Результат
703	Неудачный перенос звонка во время мероприятий (ручной)		Пройден
704	Принятие звонка во время мероприятий		Пройден

ID теста	Название теста	Скриншоты	Результат
705	Перенос звонка вне мероприятий (ручной)		Пройден
706	Перенос звонка вне мероприятий (выключен)		Пройден

ID теста	Название теста	Скриншоты	Результат
801	Успешное планирование звонка		Пройден
802	Неудачное планирование звонка		Пройден

На основе проведенных тестов можно сделать вывод о работоспособности приложения и соответствии функций заявленным в технических требованиях.

9.4 Инструкция по сборке

Настройка сервера и базы данных

ПРИМЕЧАНИЕ 1: возможно на вашем компьютере будет создана не папка **OSPanel**, а папка **OpenServer**

1. Установить OpenServer по ссылке

https://drive.google.com/open?id=1qrWESfOJMFCoivy5DwsJn8TE_FCskQqi&authuser=m1602881%40edu.misis.ru&usp=drive_fs

2. Клонировать репозиторий с GitHub в папку

/OSPanel/domains

3. Запустить файл

/OSPanel/Open Server.exe

4. Создать нового пользователя и базу данных

4.1. В трее нажать на флажок **ПКМ** и выбрать «Запустить». При необходимости, разрешить доступ к сетям

4.2. В разделе «Дополнительно» выбрать «**SQL менеджер**»

4.3. Нажать кнопку «Создать»

4.4. Нажать кнопку «Открыть». В случае возникновения ошибки ввести пароль «**root**»

4.5. Во вкладке «Инструменты» выбрать «Управление пользователями»

4.5.1. Нажать кнопку «Добавить»

4.5.2. В «Хост» записать локальный IP-адрес компьютера

4.5.3. Задать и запомнить имя пользователя латинскими буквами

4.5.4. Задать и запомнить пароль латинскими буквами

- 4.5.5. Отметить галочкой «Глобальные разрешения»
- 4.5.6. Нажать кнопку «Сохранить»
- 4.5.7. Убедиться, что в чекбоксе «Глобальные разрешения» стоит галочка, а не квадрат
- 4.5.8. Нажать кнопку «Отмена»
- 4.6. Во вкладке «Файл» выбрать «Выполнить SQL файл»
- 4.7. Выбрать файл «**caller_create_db.sql**» Из папки
/OSPanel/domains/first_common_project/DB
- 4.8. Нажать выделенную кнопку:

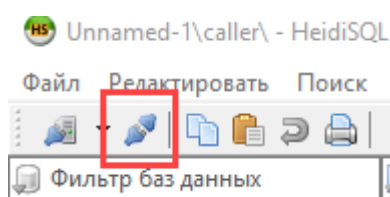


Рис. 9.4.1 – Настройка сервера

5. Настроить сервер

- 5.1. В трее нажать на флажок **ПКМ** и выбрать «**Настройки**»
- 5.2. Во вкладке «**Основные**»:
 - 5.2.1. Отметить галочкой «**Автозапуск сервера**»
 - 5.2.2. Отметить галочкой «**Требовать учётную запись администратора**»
 - 5.2.3. Нажать кнопку «**Сохранить**»
 - 5.2.4. Перезапустить OpenServer
 - 5.2.4.1. В трее нажать на флажок сервера **ПКМ**
 - 5.2.4.2. Выбрать «**Выход**»
 - 5.2.4.3. Снова запустить OpenServer (п. 3-4.1)
- 5.3. Вернуться в настройки (п. 5.1)
- 5.4. Во вкладке «**Сервер**»:
 - 5.4.1. В поле «**Настройка виртуального диска**» выбрать «**Не использовать вирт. диск**»
 - 5.4.2. В поле «**IP-адрес сервера**» указать IP-адрес компьютера

5.5. Во вкладке «Модули»:

5.5.1. Убедиться, что выбраны следующие пункты:

Модуль	Версия	Логирование
HTTP	Apache_2.4-PHP_7.2	<input checked="" type="checkbox"/> Вести лог запросов
PHP	PHP_7.3-x64	<input type="checkbox"/> Нет лога запросов
MySQL / MariaDB	MySQL-5.6-x64	<input type="checkbox"/> Вести лог запросов
PostgreSQL	Не использовать	<input type="checkbox"/> Вести лог запросов
MongoDB	Не использовать	<input type="checkbox"/> Вести лог запросов
Memcached	Не использовать	<input type="checkbox"/> Вести лог запросов
Redis	Не использовать	<input type="checkbox"/> Нет лога запросов
DNS	Не использовать	<input type="checkbox"/> Вести лог запросов

Информация
Внимание! Файлы конфигурации индивидуальны для каждого модуля. Будьте внимательны при переключении версий модулей и отдельно редактируйте настройки каждого из них.

Рис. 9.4.2 – Настройка модулей сервера

5.6. Во вкладке «Домены»:

5.6.1. В поле «Управление доменами» выбрать «Ручное управление»

5.6.2. В поле «Имя домена» написать «**project-x**»

5.6.3. В поле «Папка домена» указать путь до
/OSPanel/domains/first_common_project/SRC/Project_X_Server

5.6.4. Нажать кнопку «Добавить»

5.7. Во вкладке «Алиасы»:

5.7.1. В поле «Исходный домен» указать IP-адрес компьютера

5.7.2. В поле «Конечный домен» выбрать «**project-x**»

5.7.3. Нажать кнопку «Добавить»

5.8. Нажать кнопку «Сохранить»

6. Перезапустить OpenServer (п. 5.2.4)

Настройка Android Studio и эмулятора

ПРИМЕЧАНИЕ 2: для компьютеров на базе процессоров **AMD** могут быть необходимы дополнительные манипуляции

7. Установить Android Studio с официального сайта:

<https://developer.android.google.cn/studio>

8. Открыть проект ProjectX и настроить среду Android Studio

8.1. Нажать «Open an existing project...»

8.2. Указать путь до папки

/OSPanel/domains/first_common_project/SRC/Project_X

8.3. Принять/обновить/сконфигурировать все предложенные файлы/обновления

8.4. В разделе

Tools > SDK Manager

Убедиться, чтобы скачаны и установлены следующие пакеты:

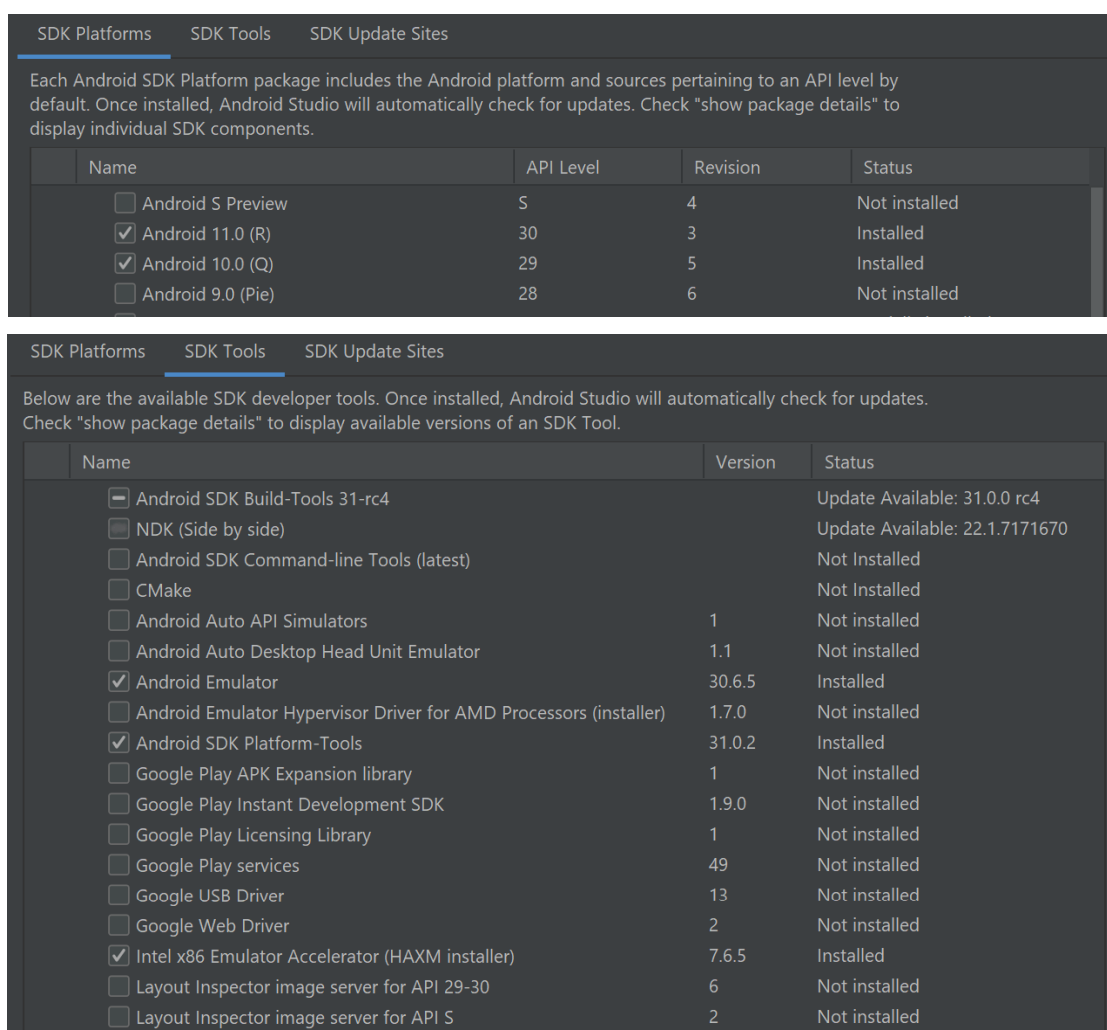


Рис. 9.4.3 – Настройка SDK на AndroidStudio

9. Прописать реквизиты сервера в приложении

9.1. Слева, в дереве проекта (вид «Android») открыть класс

app/java/com.shilina.project_x/ServerHandler

9.2.Изменить значение константы HOST на IP-адрес компьютера

9.3.Изменить значение константы DB_LOGIN на имя пользователя из п. 4.5.2

9.4.Изменить значение константы DB_PASS на пароль из п. 4.5.3

10.Нажать выделенную кнопку:

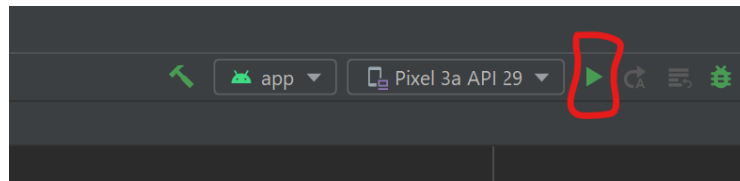


Рис. 9.4.4 – Запуск эмулятора

9.5 Программное обеспечение для командной работы

После формирования группы понадобилась платформа для взаимодействий и совместной работы над проектом.

GitHab – основная платформа для разработки, рисунок 9.5.1. Удобный и гибкий интерфейс, отслеживание изменений, простой доступ к коду делают ее наиболее актуальной для командной работы над проектом. Каждый из членов команды получил доступ к проекту, что позволило наиболее просто и эффективно участвовать в процессе и иметь актуальную версию приложения благодаря интеграции с программным обеспечением AndroidStudio, используемой для создания мобильного приложения.

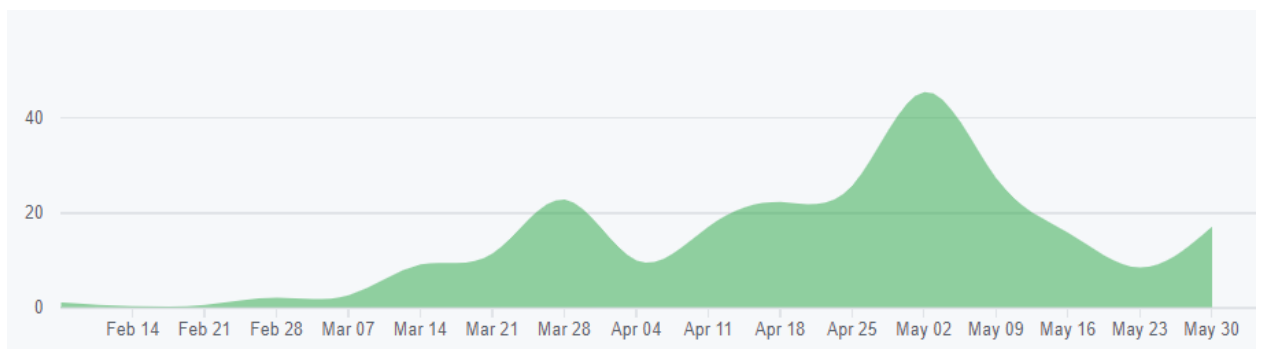


Рис. 9.5.1 – Активность команды на платформе GitHub

Основной платформой для коммуникации выступил мессенджер Telegram с созданным для команды чатом, рисунок 9.5.2. В нем

поддерживается актуальная информация по проекту, обсуждаются планы его реализации и решаются незначительные проблемы.

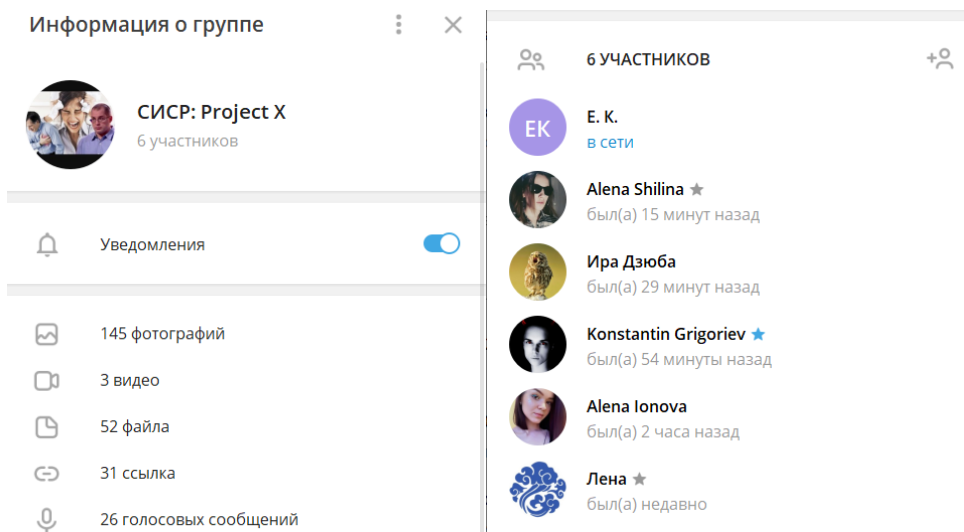


Рис. 9.5.2 – Информация о чате команды в Telegram

Голосовой канал в Discord, рисунок 9.5.3, послужил дополнением к текстовому общению, возможность стрима и удобство платформы позволили организовать среду для еженедельных созвонов команды с формированием текущих планов и решением возникающих проблем или демонстрированием успешной реализации элементов приложения.

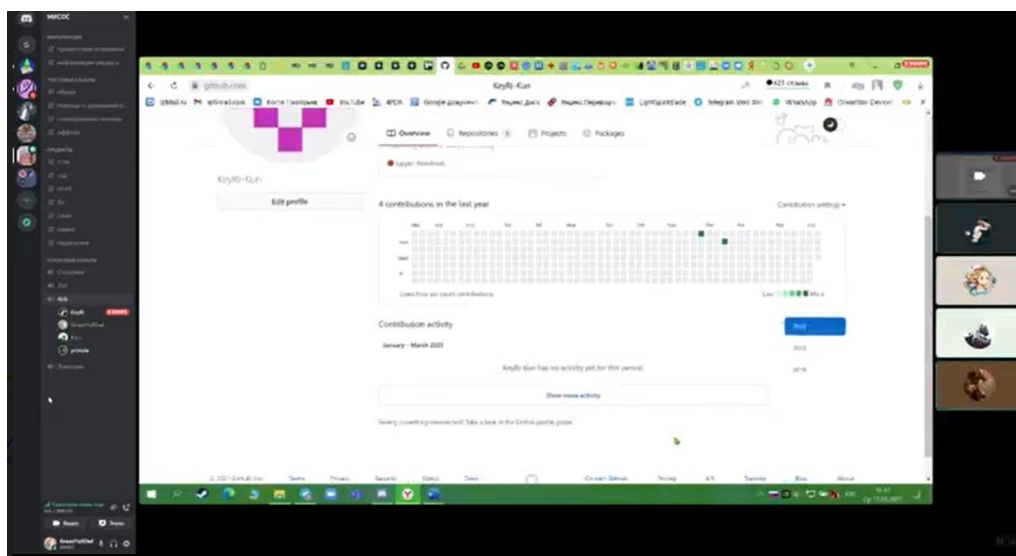


Рис. 9.5.3 – Созвон в Discord и демонстрация экрана

Для записей еженедельных созвонов и временной документации создана папка на Google Drive, 9.5.4, что позволило хранить на GitHub информацию по проекту без внутренней документации, например, текстовых файлов с изложением информации с созвонов.

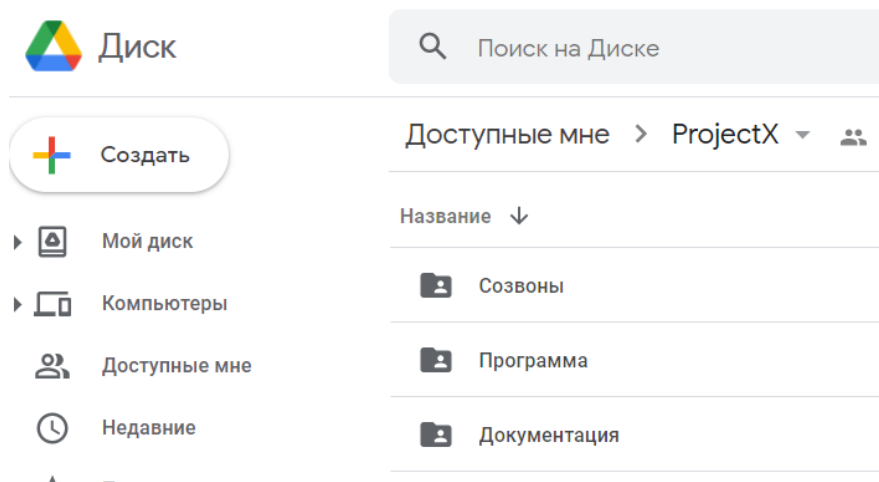


Рис. 9.5.4 – Структура папок в Google Drive

Основной платформой планирование и контроля исполнения текущей деятельности является доска Trello, рисунок 9.5.5. Графический юзерфрендли интерфейс платформы обеспечивает возможность создания заданий-карточек с назначением ответственного за выполнение, сроков, чек-листов. Использование Trello позволило сформировать желаемый календарный план работы над проектом и оценивать изменение длительности того или иного этапа и влияния его на всю работу.

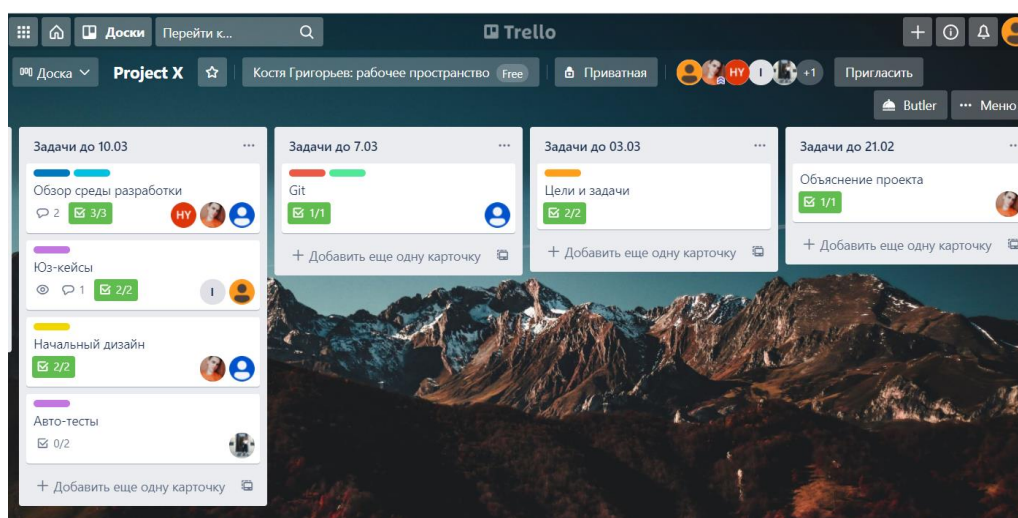


Рис. 9.5.5 – Рабочее пространство Trello

ЗАКЛЮЧЕНИЕ

В рамках выполнения курсовой работы команда студентов группы МПИ-20-4-2 познакомилась со всеми этапами разработки проекта от формирования начальной идеи и технического задания на проект и до его реализации в виде готового мобильного приложения с требуемым функционалом и характеристиками. Постоянное взаимодействие всех членов команды разработки и поддержание актуальной документации поспособствовало реализации целей и задач проекта мобильного приложения.

В целом мобильное приложение ProjectX может стать основой для дальнейшей разработки дополнительного функционала, вплоть до превращения в аналог социальной сети, когда время для созвона приложение будет предлагать не только на основе занятости пользователя, но и в соответствии со свободным временем его собеседника, так же имеющего установленное приложение ProjectX.

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ Р ИСО/МЭК 15288–2005 «Процессы жизненного цикла систем».
2. ГОСТ 19.201-78. «Единая система программной документации. Техническое задание. Требования к содержанию и оформлению»;
3. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы;
4. . Systems Engineering Fundamentals: Supplementary text prepared by the defense acquisition university press Fort Belvoir. – Virginia 22060-5565, 2001.
5. Халл Элизабет, Джексон Кен, Дик Джереми. Разработка и управление требованиями, 2005 г.
6. Карл Вегерс «Разработка требований к программному обеспечению»
7. Алистера Коберн "Современные методы описания функциональных требований к системам "
8. Ларман К. Применение UML и шаблонов проектирования/пер. с англ. Шелестова А. Ю.. - М.: Вильямс, 2017. - 727 с
9. Перерва А. Д., Иванова В. А. «Путь аналитика. Практическое руководство IT-специалиста». 2017 г. 2-е издание.
10. Халл Элизабет, Джексон Кен, Дик Джереми. Разработка и управление требованиями, 2005 г.
11. Карл Вегерс «Разработка требований к программному обеспечению»
12. Алистера Коберн "Современные методы описания функциональных требований к системам "
13. Cimperman, Rob (2006). UAT Defined: A Guide to Practical User Acceptance Testing. Pearson Education. pp. Chapter 2. ISBN 9780132702621
14. Radin, Beryl A. (2000). Beyond Machiavelli : policy analysis comes of age. Washington, D.C.: Georgetown University Press. ISBN 0878407731. OCLC 41834855
15. Alexander, Ian, and Beus-Dukic, Ljerka. Discovering Requirements: How to Specify Products and Services. Wiley, 2009.
16. Schneider, Geri and Winters, Jason P. Applying Use Cases 2nd Edition: A Practical Guide. Addison-Wesley, 2001.
17. Лекции по системному анализу // [Электронный ресурс] URL: <http://victor-safronov.narod.ru/systems-analysis/lectures/rodionov/09.html> (дата обращения: 20.03.2021)

ПРИЛОЖЕНИЕ А. ЛИСТИНГ КОДА FRONTEND-РАЗРАБОТКИ

Файлы .xml в папке SRC\Project_X\app\src\main\res\layout

activity_about_us

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/drawer_layout"
    tools:context=".AboutUsActivity"
    android:background="?attr/colorBackground">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:background="?attr/colorBackground">

        <include
            layout="@layout/main_toolbar"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/autors"
            android:text="@string/aboutUs"
            android:textSize="20sp"
            android:layout_marginStart="30dp"
            android:layout_marginEnd="30dp"
            android:textColor="?attr/colorText"
            android:background="?attr/colorBackground"/>

    </LinearLayout>

    <RelativeLayout
        android:layout_width="300dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:background="@android:color/white">

        <include
            layout="@layout/main_nav_drawer"/>

    </RelativeLayout>

</androidx.drawerlayout.widget.DrawerLayout>
```

activity_help

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/drawer_layout"
    tools:context=".HelpActivity">
```

```

        android:background="?attr/colorBackground">

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

            <include
                layout="@layout/main_toolbar"/>

            <ScrollView
                android:layout_width="match_parent"
                android:layout_height="match_parent">

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

                    <TextView
                        android:id="@+id/textHelp"
                        android:layout_width="match_parent"
                        android:layout_height="0dp"
                        android:layout_weight="4"
                        android:layout_marginHorizontal="25dp"
                        android:text="@string/help"
                        android:textColor="?attr/colorText"
                        android:textSize="20sp"
                        android:textStyle="bold" />

                    </LinearLayout>
                </ScrollView>

            </LinearLayout>

            <RelativeLayout
                android:layout_width="300dp"
                android:layout_height="match_parent"
                android:layout_gravity="start"
                android:background="@android:color/white">

                <include
                    layout="@layout/main_nav_drawer"/>

            </RelativeLayout>

        </androidx.drawerlayout.widget.DrawerLayout>

```

activity_later_calls

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/drawer_layout"
    tools:context=".LaterCallsActivity"
    android:background="?attr/colorBackground">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"

```

```

        android:background="?attr/colorBackground">

        <include
            layout="@layout/main_toolbar"/>

        <ScrollView
            android:id="@+id/scroll_view"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            <LinearLayout
                android:id="@+id/scroll_view_layout"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical" >
                </LinearLayout>
            </ScrollView>
        </LinearLayout>

        <RelativeLayout
            android:layout_width="300dp"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            android:background="@android:color/white">

            <include
                layout="@layout/main_nav_drawer"/>

        </RelativeLayout>

    </androidx.drawerlayout.widget.DrawerLayout>

```

activity_login

```

<?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"
    tools:context=".LoginActivity"
    tools:ignore="ExtraText"
    android:background="?attr/colorBackground">

    <ImageView
        android:id="@+id/logo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:src="?attr/logoSrc"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_weight="1"
    />

    <EditText
        android:id="@+id/login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
    >

```



```

        android:inputType="textPersonName"
        android:hint="Имя"
        app:layout_constraintTop_toBottomOf="@id/logo"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginTop="20dp"/>

<EditText
    android:id="@+id/password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:hint="Пароль"
    android:ems="10"
    android:inputType="textPassword"
    app:layout_constraintTop_toBottomOf="@id/login"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="20dp"/>

<Button
    android:id="@+id/sign_in"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:text="@string/sign_in"
    app:layout_constraintEnd_toEndOf="@+id/sign_up"
    app:layout_constraintStart_toStartOf="@+id/sign_up"
    android:layout_gravity="center"
    app:layout_constraintTop_toBottomOf="@+id/password"
    app:layout_constraintBottom_toBottomOf="@+id/sign_up"
    android:onClick="CheckLogin"
    android:background="?attr/colorButton"
    android:textColor="?attr/colorTextOnButton"/>

<Button
    android:id="@+id/sign_up"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:text="@string/sign_up"
    app:layout_constraintTop_toBottomOf="@id/sign_in"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    android:onClick="GoSignUp"
    android:background="?attr/colorButton"
    android:textColor="?attr/colorTextOnButton"/>

```

</androidx.constraintlayout.widget.ConstraintLayout>

activity_pay

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

```

```

        android:orientation="vertical"
    >

    <EditText
        android:id="@+id/pay_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:textSize="20sp"
        android:ems="15"
        android:maxLength="16"
        android:hint="Homep карты"
        android:inputType="number"
    />

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

        <EditText
            android:id="@+id/pay_month"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_weight="1"
            android:ems="10"
            android:maxLength="2"
            android:hint="Месяц"
            android:inputType="number"
            android:layout_marginTop="10dp"
        />

        <EditText
            android:id="@+id/pay_year"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_weight="1"
            android:ems="10"
            android:maxLength="2"
            android:hint="Год"
            android:inputType="number"
            tools:layout_editor_absoluteX="16dp"
            tools:layout_editor_absoluteY="161dp"
        />

        <EditText
            android:id="@+id/pay_cvc"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_weight="1"
            android:ems="10"
            android:hint="CVC"
            android:maxLength="3"
            android:inputType="number"
            tools:layout_editor_absoluteX="16dp"

```

```

        tools:layout_editor_absoluteY="161dp"/>

</LinearLayout>

<TextView
    android:id="@+id/textAlert"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="Введите корректные данные"
    android:textSize="15sp"
    android:textColor="#ff0000"
    android:visibility="invisible"/>

<EditText
    android:id="@+id/pay_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:gravity="center"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="16dp"
    android:textSize="20sp"
    android:ems="15"
    android:hint="ИМЯ ФАМИЛИЯ"
    android:inputType="textCapCharacters"
/>

<Button
    android:id="@+id/buy"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="?attr/colorButton"
    android:onClick="onSingUpButtonClick"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:text="КупИТЬ"
    android:textColor="?attr/colorTextOnButton"
    android:layout_marginVertical="10dp"/>

<Button
    android:id="@+id/pay_cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="?attr/colorButton"
    android:onClick="onSingUpButtonClick"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:text="Отменить"
    android:textColor="?attr/colorTextOnButton"
    android:layout_marginVertical="10dp" />

</LinearLayout>

```

activity_permissions

```

<?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"
tools:context=".PermissionsActivity"
android:background="?attr/colorBackground">

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

    <TextView
        android:id="@+id/name_perm"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingStart="20dp"
        android:paddingEnd="20dp"
        android:text="Пожалуйста, включите все разрешения. Это нужно для
стабильной работы программы"
        android:gravity="center"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:textSize="25sp"
        android:textColor="@color/sharkGrey"/>

    <Switch
        android:id="@+id/switch1"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:padding="12dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:text="@string/perm1"
        android:onClick="onPermClick"
        tools:ignore="UseSwitchCompatOrMaterialXml"
        android:textColor="?attr/colorText"/>

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_marginHorizontal="20dp"
        android:layout_gravity="center"
        android:text="@string/permText"
        android:textColor="?attr/colorText"/>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="20dp"
        android:paddingEnd="20dp"
        android:layout_gravity="center"
        android:layout_marginVertical="5dp"
        android:text="@string/perm2"
        android:onClick="onOverClick"
        />

    <Button
        android:id="@+id/button2"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="20dp"
        android:paddingEnd="20dp"
        android:layout_gravity="center"
        android:text="@string/perm3"
        android:onClick="onBackClick"/>

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:layout_gravity="center"
    android:layout_marginVertical="5dp"
    android:text="@string/perm4"
    android:onClick="onBatClick"/>

<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:layout_gravity="center"
    android:text="@string/perm5"
    android:onClick="onAutoClick"/>

</LinearLayout>

<Button
    android:id="@+id/buttonPerm"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="15sp"

    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:layout_gravity="center"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="50dp"
    android:text="@string/accept"
    android:onClick="onContinueClick"
    android:background="?attr/colorButton"
    android:textColor="?attr/colorTextOnButton"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

activity_settings

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/drawer_layout"

```

```

tools:context=".SettingsActivity"
android:background="?attr/colorBackground">

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

    <include
        layout="@layout/main_toolbar"/>

    <LinearLayout
        android:id="@+id/theme"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/textTheme"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:layout_constraintTop_toBottomOf="@layout/main_toolbar"
            android:layout_marginLeft="50dp"
            android:text="Тема"
            android:textSize="14dp"
            android:textColor="?attr/colorText"/>
        <Spinner
            android:id="@+id/spinnerTheme"
            android:layout_width="80pt"
            android:layout_height="wrap_content"
            android:layout_marginLeft="100dp"
            android:layout_marginRight="100dp" />
    </LinearLayout>

    <TextView
        android:id="@+id/textModeIn"
        app:layout_constraintTop_toBottomOf="@id/theme"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:layout_marginLeft="50dp"
        android:layout_marginVertical="5dp"
        android:text="Переносить во время мероприятий:"
        android:textSize="14dp"
        android:textColor="?attr/colorText"/>
    <RadioGroup
        android:id="@+id/modeIn"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_marginLeft="50dp"
        android:layout_weight="2"
        app:layout_constraintTop_toBottomOf="@id/textModeIn">
        <RadioButton
            android:id="@+id/modeInHand"
            android:layout_width="wrap_content"
            android:layout_height="25dp"
            android:text="Вручную"
            android:textColor="?attr/colorText"/>
        <RadioButton
            android:id="@+id/modeInAuto"
            android:layout_width="wrap_content"
            android:layout_height="25dp"
            android:text="Авто (на свободное время)"

```

```

        android:checked="true"
        android:textColor="?attr/colorText"/>
</RadioGroup>

<TextView
    android:id="@+id/textModeOut"
    app:layout_constraintTop_toBottomOf="@id/theme"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:layout_marginLeft="50dp"
    android:text="Переносить вне мероприятий:"
    android:textSize="14dp"
    android:layout_marginVertical="5dp"
    android:textColor="?attr/colorText"/>
<RadioGroup
    android:id="@+id/modeOut"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_marginLeft="50dp"
    android:layout_weight="2"
    app:layout_constraintTop_toBottomOf="@id/textModeOut">
    <RadioButton
        android:id="@+id/modeOutHand"
        android:layout_width="wrap_content"
        android:layout_height="25dp"
        android:text="Вручную"
        android:textColor="?attr/colorText"/>
    <RadioButton
        android:id="@+id/modeOutOff"
        android:layout_width="wrap_content"
        android:layout_height="25dp"
        android:text="ВЫКЛЮЧИТЬ"
        android:checked="true"
        android:textColor="?attr/colorText"/>
</RadioGroup>

<EditText
    android:id="@+id/settings_password"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_weight="1.5"
    android:ems="10"
    android:hint="Пароль"
    android:inputType="textPassword"
    android:layout_gravity="center_horizontal"
    android:layout_marginVertical="5dp"
    android:textSize="14dp"
    android:textColor="?attr/colorText"
    android:textColorHint="?attr/colorText"/>

<EditText
    android:id="@+id/settings_password_check"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_weight="1.5"
    android:ems="10"
    android:hint="Повторите пароль"
    android:inputType="textPassword"
    android:layout_gravity="center_horizontal"
    android:layout_marginVertical="5dp"
    android:textSize="14dp"
    android:textColor="?attr/colorText"
    android:textColorHint="?attr/colorText"/>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="6"
    android:orientation="vertical">
    <Button
        android:id="@+id/password_change_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="20dp"
        android:paddingEnd="20dp"
        android:layout_gravity="center"
        android:layout_marginVertical="5dp"
        android:onClick="onPasswordChangeClick"
        android:text="Сменить пароль"
        android:background="?attr/colorButton"
        android:textColor="?attr/colorTextOnButton"/>
    <Button
        android:id="@+id/reset_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="20dp"
        android:paddingEnd="20dp"
        android:layout_gravity="center"
        android:layout_marginVertical="5dp"
        android:onClick="onResetClick"
        android:text="Сбросить настройки"
        android:background="?attr/colorButton"
        android:textColor="?attr/colorTextOnButton"/>
    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingStart="20dp"
        android:paddingEnd="20dp"
        android:layout_gravity="center"
        android:layout_marginVertical="5dp"
        android:onClick="onPermissionsClick"
        android:text="Посмотреть разрешения"
        android:background="?attr/colorButton"
        android:textColor="?attr/colorTextOnButton"/>
    </LinearLayout>
</LinearLayout>

<RelativeLayout
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:background="@android:color/white">

    <include
        layout="@layout/main_nav_drawer"/>

</RelativeLayout>

</androidx.drawerlayout.widget.DrawerLayout>

```

activity_sign_up

```

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

```



```

android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SignUpActivity"
android:orientation="vertical"
android:background="?attr/colorBackground">

```

```

<EditText
    android:id="@+id/email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="30dp"
    android:ems="10"
    android:hint="Google почта"
    android:inputType="textPersonName" />

```

```

<EditText
    android:id="@+id/phone_number"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="30dp"
    android:ems="10"
    android:maxLength="11"
    android:hint="Номер телефона"
    android:inputType="textPersonName" />

```

```

<EditText
    android:id="@+id/enterPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="30dp"
    android:ems="10"
    android:hint="Пароль"
    android:inputType="textPassword" />

```

```

<EditText
    android:id="@+id/checkPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="30dp"
    android:ems="10"
    android:hint="Повторите пароль"
    android:inputType="textPassword" />

```

```

<Button
    android:id="@+id/signUpNow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:text="Зарегистрироваться"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="70dp"
    android:onClick="onSignUpButtonClick"
    android:background="?attr/colorButton"
    android:textColor="?attr/colorTextOnButton"/>

```

```

<TextView

```

```

        android:id="@+id/textAlertSign"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginVertical="30dp"
        android:text="Введите корректные данные"
        android:textSize="15sp"
        android:textColor="#ff0000"
        android:visibility="invisible"/>
    </LinearLayout>

```

activity_status

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/drawer_layout"
    tools:context=".StatusActivity"
    android:background="?attr/colorBackground">

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

        <include
            layout="@layout/main_toolbar"/>

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

            <TextView
                android:id="@+id/textStatus"
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="4"
                android:gravity="center"
                android:text="@string/statusNo"
                android:textSize="30sp"
                android:textStyle="bold"
                android:textColor="?attr/colorText"/>

            <Button
                android:id="@+id/button"
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:paddingStart="20dp"
                android:paddingEnd="20dp"
                android:layout_margin="50dp"
                android:layout_weight="1"
                android:onClick="onPurchaseClick"
                android:text="Купить премиум за 250p."
                app:layout_constraintTop_toBottomOf="@id/textStatus"
                android:background="?attr/colorButton"
                android:textColor="?attr/colorTextOnButton"/>

        </LinearLayout>
    </LinearLayout>

```

```

</LinearLayout>

<RelativeLayout
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:background="@android:color/white">

    <include
        layout="@layout/main_nav_drawer"/>

</RelativeLayout>

</androidx.drawerlayout.widget.DrawerLayout>

```

activity_status_prem

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/drawer_layout"
    android:background="?attr/colorBackground">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <include
            layout="@layout/main_toolbar"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="@string/statusYes"
            android:textSize="30sp"
            android:textStyle="bold"
            android:gravity="center"
            android:layout_marginHorizontal="20dp"
            android:textColor="?attr/colorText"/>

    </LinearLayout>

    <RelativeLayout
        android:layout_width="300dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:background="@android:color/white">

        <include
            layout="@layout/main_nav_drawer"/>

    </RelativeLayout>

</androidx.drawerlayout.widget.DrawerLayout>

```

call_logo_layout

```

<?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"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/call_logo_layout">

<ImageView
    android:id="@+id/call_logo_button"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="?attr/logoSrc"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

main_nav_drawer

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="?attr/colorDrawerBody"
    >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:layout_margin="16dp"
        android:onClick="onLogoClick"
        android:src="?attr/logoSrc" />

    <TextView
        android:id="@+id/nickname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="?attr/colorText"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:background="?attr/colorDrawerBody"
        />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:background="?attr/colorDrawerBody">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center_vertical"

```

```

        android:onClick="onLaterActClick"
        android:orientation="horizontal">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_weight="1"
            android:padding="12dp"
            android:text="Планирование звонков"
            android:textStyle="bold"
            android:textColor="?attr/colorDrawerTextBody"/>

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="48dp"
            android:src="@drawable/ic_later_call" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:onClick="onStatusActClick"
        android:orientation="horizontal">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_weight="1"
            android:padding="12dp"
            android:text="Мой статус"
            android:textStyle="bold"
            android:textColor="?attr/colorDrawerTextBody"/>

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="48dp"
            android:src="@drawable/ic_status" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:onClick="onSettingsActClick"
        android:orientation="horizontal">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_weight="1"
            android:padding="12dp"
            android:text="Настройки"
            android:textStyle="bold"
            android:textColor="?attr/colorDrawerTextBody"/>

        <ImageView
            android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_marginEnd="48dp"
        android:src="@drawable/ic_settings" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:onClick="onHelpActClick"
    android:orientation="horizontal">

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_weight="1"
        android:padding="12dp"
        android:text="Помощь"
        android:textStyle="bold"
        android:textColor="?attr/colorDrawerTextBody"/>

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="48dp"
        android:src="@drawable/ic_help" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:onClick="onAboutUsActClick"
    android:orientation="horizontal">

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_weight="1"
        android:padding="12dp"
        android:text="О разработчиках"
        android:textStyle="bold"
        android:textColor="?attr/colorDrawerTextBody"/>

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="48dp"
        android:src="@drawable/ic_about_us" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:onClick="onLogoutClick"
    android:orientation="horizontal">

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"

```

```

        android:layout_weight="1"
        android:padding="12dp"
        android:text="Выйти"
        android:textStyle="bold"
        android:textColor="?attr/colorDrawerTextBody"/>

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="48dp"
            android:src="@drawable/ic_logout" />
    </LinearLayout>

</LinearLayout>

</LinearLayout>

```

main_toolbar

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="12dp"
    android:gravity="center_vertical"
    android:background="?attr/colorBackground">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onMenuClick"
        android:src="@drawable/ic_menu"/>

    <TextView
        android:id="@+id/nameOfWindow"
        android:layout_width="0dp"
        android:layout_height="40dp"
        android:layout_weight="35"
        android:text="@string/nameOfWindow"
        android:textAlignment="center"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="?attr/colorText"/>

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="32dp"
        android:layout_height="32dp"
        android:layout_weight="1"
        android:src="?attr/logoSrc" />
</LinearLayout>

```

one_call_view

```

<?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"
    android:id="@+id/one_call_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

```

```

        android:padding="5sp">

        <TextView
            android:id="@+id/one_call_view_text"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:background="@drawable/one_call_view_text_back"
            android:gravity="center_horizontal"
            android:padding="5sp"
            android:textColor="#ffffff"
            android:textSize="12sp"
            app:layout_constraintHeight_percent="0.5"
            app:layout_constraintHorizontal_chainStyle="packed"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toLeftOf="@id/one_call_view_button"
            app:layout_constraintWidth_percent="0.75" />

        <Button
            android:id="@+id/one_call_view_button"
            android:layout_width="0dp"
            android:layout_height="0dp"
            android:background="@drawable/one_call_view_button_back"
            android:gravity="center"
            android:padding="2sp"
            android:text="Отменить"
            android:textColor="#ffffff"
            android:textSize="10sp"
            app:layout_constraintHeight_percent="1"
            app:layout_constraintHorizontal_bias="0"
            app:layout_constraintLeft_toRightOf="@id/one_call_view_text"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="@id/one_call_view_text"
            app:layout_constraintBottom_toBottomOf="@id/one_call_view_text"
            app:layout_constraintWidth_percent="0.25" />

    </androidx.constraintlayout.widget.ConstraintLayout>

```

plan_call_layout

```

<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/plan_call_layout"
    android:paddingHorizontal="15dp"
    android:paddingTop="160dp"
    android:paddingBottom="250dp"
    android:background="#202020"
    >

    <LinearLayout
        android:id="@+id/bubble_left"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:background="@drawable/one_call_view_text_back"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_chainStyle="packed"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toLeftOf="@id/bubble_right"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintWidth_percent="0.75">

```



```

<TextView
    android:id="@+id/bubble_text"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="2"
    android:hint="Type text"
    android:text="@string/textSMS"
    android:gravity="center"
    android:paddingTop="5sp"
    android:textColor="#ffffff"
    android:textSize="15sp" />
<TimePicker
    android:id="@+id/bubble_time_picker"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="6"
    android:amPmTextColor="#FFFFFF"
    android:background="#00FFFFFF"
    android:timePickerMode="spinner"
    android:numbersTextColor="?attr/colorTextOnButton"/>
<DatePicker
    android:id="@+id/bubble_date_picker"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="6"
    android:amPmTextColor="#FFFFFF"
    android:background="#00FFFFFF"
    android:datePickerMode="spinner"
    android:calendarViewShown="false"
    android:numbersTextColor="?attr/colorTextOnButton"/>
<TextView
    android:id="@+id/bubble_sign"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="center"
    android:text="Информация из вашего календаря"
    android:textColor="#c3d7e6"
    android:textSize="13sp" />
<EditText
    android:id="@+id/bubble_phone"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="2"
    android:gravity="center"
    android:textColor="#c3d7e6"
    android:textSize="15sp"
    android:hint="Номер телефона собеседника"
    android:maxLines="1"
    android:lines="1"
    android:singleLine="true"
    android:inputType="number"
    android:numbersTextColor="?attr/colorTextOnButton"/>
</LinearLayout>

<LinearLayout
    android:id="@+id/bubble_right"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="@id/bubble_left"
    app:layout_constraintLeft_toRightOf="@id/bubble_left"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="@id/bubble_left"

```

```

        app:layout_constraintWidth_percent="0.25">
<Button
    android:id="@+id/bubble_button_send"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="@drawable/one_call_view_button_back"
    android:gravity="center"
    android:text="Далее"
    android:textColor="#ffffff"
    android:textSize="12sp"
    app:layout_constraintWidth_percent="0.25" />
<Button
    android:id="@+id/bubble_button_wait15"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="@drawable/one_call_view_button_back"
    android:gravity="center"
    android:text="+ 15 минут"
    android:textColor="#ffffff"
    android:textSize="12sp"
    app:layout_constraintWidth_percent="0.25" />
<Button
    android:id="@+id/bubble_button_wait30"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="@drawable/one_call_view_button_back"
    android:gravity="center"
    android:text="+ 30 минут"
    android:textColor="#ffffff"
    android:textSize="12sp"
    app:layout_constraintWidth_percent="0.25" />
<Button
    android:id="@+id/bubble_button_wait1"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="@drawable/one_call_view_button_back"
    android:gravity="center"
    android:text="+ 1 час"
    android:textColor="#ffffff"
    android:textSize="12sp" />
<Button
    android:id="@+id/bubble_button_return"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="@drawable/one_call_view_button_back"
    android:gravity="center"
    android:text="Отменить"
    android:textColor="#ffffff"
    android:textSize="12sp" />
</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

text_bubbl

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"

```

```

        android:layout_height="match_parent"
        xmlns:app="http://schemas.android.com/apk/res-auto"
    >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:id="@+id/text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Файлы .xml в папке SRC\Project_X\app\src\main\res\values

Colors

```

<?xml version="1.0" encoding="utf-8"?>
<resources>

    <color name="sharkPrimary">#1455af</color>
    <color name="sharkHead">#0e4187</color>
    <color name="sharkGrey">#666666</color>
    <color name="sharkWhite">#FFFFFF</color>

    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>

    <color name="panteraPrimary">#212121</color>
    <color name="panteraHead">#000000</color>
    <color name="panteraGrey">#424242</color>
    <color name="panteraWhite">#FFFFFF</color>
    <color name="panteraDrawer">#666666</color>
    <color name="panteraBack">#303030</color>
    <color name="panteraText">#a0a0a0</color>

    <color name="enotPrimary">#785A46</color>
    <color name="enotHead">#4B311E</color>
    <color name="enotGrey">#666666</color>
    <color name="enotWhite">#FFFFFF</color>
    <color name="enotDrawer">#FFFFFF</color>
    <color name="enotBright">#9C4B00</color>
    <color name="enot2">#D37830</color>

</resources>

```

Strings

```

<resources>
    <string name="app_name">Project_X</string>
    <string name="permissions">Permissions</string>
    <string name="perm1">Системные разрешения</string>
    <string name="perm2">Включать поверх приложений</string>
    <string name="perm3">Разрешить в фоновом режиме</string>
    <string name="perm4">Отключить энергосбережение</string>
    <string name="perm5">И здесь отключить энергосбережение</string>
    <string name="permText">Дополнительные разрешения:</string>
    <string name="accept">Продолжить</string>
    <string name="alarmAccept">Необходимо принять все разрешения</string>
    <string name="alarmAcceptText">Проверьте еще раз</string>

```

```

<string name="sign_in">Войти</string>
<string name="sign_up">Зарегистрироваться</string>
<string name="nameOfWindow">%1$s</string>
<string name="aboutUs">Шилина
Алена\nhhttps://github.com/ShilinaAle\n\nГригорьев
Константин\nhhttps://github.com/KeyRi-Kun\n\nДзюба
Ирина\nhhttps://github.com/dzyubaira\n\nКорлякова
Екатерина\nhhttps://github.com/Prostotarakan\n\nЮрасова
Елена\nhhttps://github.com/greenyellowl\n\nИонова
Алена\nhhttps://github.com/ionova0070</string>
<string-array name="themes">
    <item>Акула</item>
    <item>Енот</item>
    <item>Пантера</item>
</string-array>
<string name="statusNo">Ваш статус - обычный\n\nА можно получить красивые
темы всего за 250р :3</string>
<string name="statusYes">Ваш статус - Премиум\n\nСпасибо, что Вы с
нами!</string>
<string name="textSMS">Я хочу запланировать с вами звонок: %1$s</string>
<string name="cancelSMS">Простите, я не смогу присутствовать на звонке,
запланированном на: %1$s</string>
<string name="textCalendar">Этот звонок был создан %2$s.\nНомер
собеседника: %1$s</string>
<string name="textOneCall">Звонок с абонентом: %1$s\nБыл запланирован:
%2$s\nБудет: %3$s</string>
<string name="help">
    Приложение ProjectX упрощает управление звонками на вашем
    смартфоне.\n\n\nОсновные режимы работы:\n\n
    - Автоматический: поступающий звонок не отвлекает вас от мероприятий,
    записанных в календаре: собеседнику отправляется сообщение с предложением
    перезвонить в ближайшее свободное время.\n\n
    - Ручной: предлагает при поступающем звонке перенести его на
    ближайшее свободное время (в соответствии с мероприятиями в календаре) или на
    любое удобное для вас время.\n\nПри планировании/отмене звонков - собеседник
    получает SMS-уведомление, а в календаре заносится/удаляется событие.\n\nВыбор
    тем Акулы, Енота или Черной пантеры, доступный для пользователей премиум-
    аккаунта, сохранит ваш стиль даже в мелочах.</string>
</resources>

```

Styles

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <attr name="logoSrc" format="reference" />

    <attr name="colorText" format="color"/>
    <attr name="colorButton" format="color"/>
    <attr name="colorTextOnButton" format="color"/>
    <attr name="colorBackground" format="color"/>
    <attr name="colorDrawerTextHead" format="color"/>
    <attr name="colorDrawerBody" format="color"/>
    <attr name="colorDrawerTextBody" format="color"/>

    <attr name="onGradientStart" format="color"/>
    <attr name="onGradientEnd" format="color"/>
    <attr name="offGradientStart" format="color"/>
    <attr name="offGradientEnd" format="color"/>
</resources>

```

Themes

```

<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme - shark -->

```

```

<style name="Theme.Project_X"
parent="Theme.AppCompat.DayNight.DarkActionBar">

    <item name="colorPrimary">@color/sharkPrimary</item>
    <item name="colorPrimaryVariant">@color/sharkHead</item>
    <item name="colorText">@color/sharkGrey</item>
    <item name="colorButton">@color/sharkPrimary</item>
    <item name="colorTextOnButton">@color/sharkWhite</item>
    <item name="colorBackground">@color/sharkWhite</item>
    <item name="colorOnPrimary">@color/sharkWhite</item>
    <item name="colorDrawerTextHead">@color/sharkGrey</item>
    <item name="colorDrawerBody">@color/white</item>
    <item name="colorDrawerTextBody">@color/sharkPrimary</item>

    <!-- <item name="colorOnSecondary">@color/black</item>-->

    <!-- Logo and gradients. -->
    <item name="logoSrc">@drawable/logo_shark</item>
    <item name="onGradientStart">#1455af</item>
    <item name="onGradientEnd">#46a5e6</item>
    <item name="offGradientStart">#557d9b</item>
    <item name="offGradientEnd">#c3d7e6</item>
</style>

<!-- Base application theme. -->
<style name="Theme.Project_X_pantera"
parent="Theme.AppCompat.DayNight.DarkActionBar">

    <item name="colorPrimary">@color/panteraPrimary</item>
    <item name="colorPrimaryVariant">@color/panteraHead</item>
    <item name="colorText">@color/panteraDrawer</item>
    <item name="colorButton">@color/panteraPrimary</item>
    <item name="colorTextOnButton">@color/panteraWhite</item>
    <item name="colorBackground">@color/panteraBack</item>
    <item name="colorOnPrimary">@color/panteraWhite</item>
    <item name="colorDrawerTextHead">@color/panteraWhite</item>
    <item name="colorDrawerBody">@color/panteraDrawer</item>
    <item name="colorDrawerTextBody">@color/panteraPrimary</item>

    <!-- Logo and gradients. -->
    <item name="logoSrc">@drawable/logo_pantera</item>
    <item name="onGradientStart">@color/panteraGrey</item>
    <item name="onGradientEnd">@color/panteraDrawer</item>
    <item name="offGradientStart">@color/panteraDrawer</item>
    <item name="offGradientEnd">@color/panteraBack</item>

</style>

<!-- Base application theme. -->
<style name="Theme.Project_X_enot"
parent="Theme.AppCompat.DayNight.DarkActionBar">
    <!-- Primary brand color. -->

    <item name="colorPrimary">@color/enotPrimary</item>
    <item name="colorPrimaryVariant">@color/enotHead</item>
    <item name="colorText">@color/enotPrimary</item>
    <item name="colorButton">@color/enotPrimary</item>
    <item name="colorTextOnButton">@color/enotWhite</item>
    <item name="colorBackground">@color/enotWhite</item>
    <item name="colorOnPrimary">@color/enotWhite</item>
    <item name="colorDrawerTextHead">@color/enotGrey</item>
    <item name="colorDrawerBody">@color/enotDrawer</item>
    <item name="colorDrawerTextBody">@color/enotPrimary</item>

```

```
    <!-- Logo and gradients. -->
    <item name="logoSrc">@drawable/logo_enot</item>
    <item name="onGradientStart">@color/enotHead</item>
    <item name="onGradientEnd">@color/enotPrimary</item>
    <item name="offGradientStart">@color/enotBright</item>
    <item name="offGradientEnd">@color/enot2</item>

</style>

</resources>
```

ПРИЛОЖЕНИЕ Б. ЛИСТИНГ КОДА BACKEND-РАЗРАБОТКИ

Файлы .java в папке

SRC\Project_X\app\src\main\java\com\shilina\project_x

AboutUsActivity

```
package com.shilina.project_x;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;

public class AboutUsActivity extends DrawerActivity {

    public static final String className =
Thread.currentThread().getStackTrace()[2].getClassName();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SettingsActivity.chooseTheme(this);
        setContentView(R.layout.activity_about_us);
        super.drawerLayout = findViewById(R.id.drawer_layout);
        super.className = className;
        setNickname();
        Log.i("LOOK HERE: AUA", "Opened activity is: " + className);

        TextView textById = findViewById(R.id.nameOfWindow);
        textById.setText("О разработчиках");
    }
}
```

BroadcastService

```
package com.shilina.project_x;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Build;
import android.os.IBinder;
import android.util.Log;

import androidx.annotation.RequiresApi;
import androidx.core.app.NotificationCompat;

public class BroadcastService extends Service {

    private static PhoneHandler phoneCall;
```

```

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Log.i("LOOK HERE: BroadcastService", "PhoneHandler started background
running");

    phoneCall = new PhoneHandler();
    IntentFilter phoneCall_filter = new
IntentFilter("android.intent.action.PHONE_STATE");
    if (!PhoneHandler.isRegistered) {
        registerReceiver(phoneCall, phoneCall_filter);
        Log.i("LOOK HERE: BroadcastService", "Registering PhoneHandler");
        PhoneHandler.isRegistered = true;
    } else {
        Log.i("LOOK HERE: BroadcastService", "PhoneHandler is
registered");
    }

    return START_STICKY;
}

@Override
public void onCreate() {
    super.onCreate();
    Log.i("LOOK HERE: BroadcastService", "Created");

    NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
    String channelId = Build.VERSION.SDK_INT >= Build.VERSION_CODES.O ?
createNotificationChannel(notificationManager) : "";
    NotificationCompat.Builder notificationBuilder = new
NotificationCompat.Builder(this, channelId);
    Notification notification = notificationBuilder.setOngoing(true)
        .setSmallIcon(R.mipmap.ic_launcher_round)
        .setPriority(NotificationManager.IMPORTANCE_LOW)
        .setCategory(NotificationCompat.CATEGORY_SERVICE)
        .build();
    startForeground(1001, notification);

    //Toast.makeText(this, "Напоминание добавлено в календарь",
Toast.LENGTH_SHORT).show();
}

@Override
public void onDestroy() {
    super.onDestroy();
    Log.i("LOOK HERE: BroadcastService", "Destroyed");
    if (PhoneHandler.isRegistered) {
        unregisterReceiver(phoneCall);
        PhoneHandler.isRegistered = false;
        phoneCall = null;
    }
}

@RequiresApi(Build.VERSION_CODES.O)
private String createNotificationChannel (NotificationManager
notificationManager) {
    String channelId = "my_service_channelid";
    String channelName = "My Foreground Service";
    NotificationChannel channel = new NotificationChannel(channelId,
channelName, NotificationManager.IMPORTANCE_HIGH);
    channel.setImportance(NotificationManager.IMPORTANCE_NONE);
    channel.setLockscreenVisibility(Notification.VISIBILITY_PRIVATE);
    notificationManager.createNotificationChannel(channel);
}

```



```

        return channelId;
    }

    public BroadcastService() {
    }
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}

```

CalendarHandler

```

package com.shilina.project_x;

import android.Manifest;
import android.content.ContentResolver;
import android.content.ContentUris;
import android.content.ContentValues;
import android.content.Context;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Color;
import android.net.Uri;
import android.provider.CalendarContract;
import android.provider.CalendarContract.Calendars;
import android.provider.CalendarContract.Events;
import android.util.Log;

import androidx.core.app.ActivityCompat;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;

public class CalendarHandler {

    public static boolean isCalendarExists(Context context) {
        Cursor cur = null;
        ContentResolver cr = context.getContentResolver();
        Uri uriCal = Calendars.CONTENT_URI;
        uriCal = uriCal.buildUpon()
            .appendQueryParameter(CalendarContract.CALLER_IS_SYNCADAPTER,
"true")
            .appendQueryParameter(Calendars.ACCOUNT_NAME,
"cal@projectx.com")
            .appendQueryParameter(Calendars.ACCOUNT_TYPE,
"com.projectx").build();

        String[] projection = new String[] { Calendars.ACCOUNT_NAME,
Calendars.ACCOUNT_TYPE};
        cur = cr.query(uriCal, projection, null, null, null);
        while (cur.moveToNext()) {
            String calType =
cur.getString(cur.getColumnIndex(Calendars.ACCOUNT_TYPE));
            if (calType.equals("com.projectx")) {
                Log.i("LOOK HERE: CalendarHandler", "Calendar exists");
                cur.close();
                return true;
            }
        }
        cur.close();
    }
}

```

```

        return false;
    }

    public static void addCalendar(Context context) {
        if (ActivityCompat.checkSelfPermission(context,
Manifest.permission.WRITE_CALENDAR) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        if (isCalendarExists(context)) {
            return;
        }
        Log.i("LOOK HERE: CalendarHandler", "ProjectX is to be created");
        ContentResolver cr = context.getContentResolver();
        ContentValues contentValues = new ContentValues();
        contentValues.put(Calendar.ACCOUNT_NAME, "cal@projectx.com");
        contentValues.put(Calendar.ACCOUNT_TYPE, "com.projectx");
        contentValues.put(Calendar.NAME, "ProjectX Calendar");
        contentValues.put(Calendar.CALENDAR_DISPLAY_NAME, "ProjectX
Calendar");
        contentValues.put(Calendar.CALENDAR_COLOR, Color.BLUE);
        contentValues.put(Calendar.CALENDAR_ACCESS_LEVEL,
Calendar.CAL_ACCESS_OWNER);
        contentValues.put(Calendar.OWNER_ACCOUNT, "KGrigor98@gmail.com");
        contentValues.put(Calendar.ALLOWED_REMINDERS, "METHOD_ALERT,
METHOD_EMAIL, METHOD_ALARM");
        contentValues.put(Calendar.ALLOWED_ATTENDEE_TYPES, "TYPE_OPTIONAL,
TYPE_REQUIRED, TYPE_RESOURCE");
        contentValues.put(Calendar.ALLOWED_AVAILABILITY, "AVAILABILITY_BUSY,
AVAILABILITY_FREE, AVAILABILITY TENTATIVE");
        contentValues.put(Calendar.SYNC_EVENTS, 1);
        Uri.Builder builder =
CalendarContract.Calendars.CONTENT_URI.buildUpon();
        builder.appendQueryParameter(Calendar.ACCOUNT_NAME,
"cal@projectx.com");
        builder.appendQueryParameter(Calendar.ACCOUNT_TYPE,
CalendarContract.ACCOUNT_TYPE_LOCAL);
        builder.appendQueryParameter(CalendarContract.CALLER_IS_SYNCADAPTER,
"true");
        cr.insert(builder.build(), contentValues);
    }

    public static void deleteCalendar(Context context) {
        addCalendar(context);
        ContentResolver cr = context.getContentResolver();
        String calendarId = getCalendarID(context);
        Uri uri = ContentUris.withAppendedId(Calendar.CONTENT_URI,
Integer.parseInt(calendarId));
        cr.delete(uri, null, null);
    }

    public static long getFreeTimeFromCalendar(Context context, long
currentTime) {
        addCalendar(context);
        long freeTimeMil = -1;

        Cursor cur = null;
        ContentResolver cr = context.getContentResolver();
        Uri uri = Events.CONTENT_URI;

        String[] from = {
            Events.DTSTART,
            Events.DTEND,
            Events.EVENT_TIMEZONE,
            Events.ALL_DAY

```

```

    };
    String where = "(" + Events.DTSTART + " <= ?) AND (" + Events.DTEND
+ " >= ?))";
    String[] equal = new String[] { Long.toString(currentTime),
Long.toString(currentTime) };

    cur = cr.query(uri, from, where, equal, null);
    while (cur.moveToNext()) {
        long dtend = cur.getLong(cur.getColumnIndex(Events.DTEND));
        if ((dtend >= freeTimeMil) &&
(cur.getLong(cur.getColumnIndex(Events.ALL_DAY)) == 0)) {
            freeTimeMil = dtend;
        }
    }

    cur.close();

    if (freeTimeMil < 0) {
        Log.i("LOOK HERE: CalendarHandler", "No events at that time");
    } else {
        if (freeTimeMil != currentTime){
            Log.i("LOOK HERE: CalendarHandler", "Сравниваются " +
freeTimeMil + " и " + currentTime);
            freeTimeMil = getFreeTimeFromCalendar(context, freeTimeMil);
        }
        Log.i("LOOK HERE: CalendarHandler", "Current event has been
found");
    }
    return freeTimeMil;
}

public static boolean isFreeAt(Context context, long timeToCheck){
    return (getFreeTimeFromCalendar(context, timeToCheck) == -1);
}

public static String getCalendarID(Context context) {
    ContentResolver cr = context.getContentResolver();
    String calendarID = null;
    Uri uriCal = Calendars.CONTENT_URI;
    uriCal = uriCal.buildUpon()
        .appendQueryParameter(CalendarContract.CALLER_IS_SYNCADAPTER,
"true")
        .appendQueryParameter(Calendars.ACCOUNT_NAME,
"cal@projectx.com")
        .appendQueryParameter(Calendars.ACCOUNT_TYPE,
"com.projectx").build();
    String[] projection = new String[] { Calendars._ID,
Calendars.ACCOUNT_TYPE};
    Cursor curCal = null;
    curCal = cr.query(uriCal, projection, null, null, null);
    while (curCal.moveToNext()) {
        String calType =
curCal.getString(curCal.getColumnIndex(Calendars.ACCOUNT_TYPE));
        Log.i("LOOK HERE: CalendarHandler", "\nName: " + calType);
        if (calType.equals("com.projectx")) {
            calendarID =
curCal.getString(curCal.getColumnIndex(Calendars._ID));
        }
    }
    curCal.close();
    return calendarID;
}

public static void addEvent(Context context, String phoneNumber, long

```

```

callStartTime, long freeTime) {
    addCalendar(context);
    if (getEventUri(context, phoneNumber, callStartTime, freeTime) ==
null) {
        String descr =
context.getResources().getString(R.string.textCalendar, phoneNumber,
getTimeStringFromLong(callStartTime, "dd.MM.y HH:mm"));
        ContentResolver cr = context.getContentResolver();
        ContentValues values = new ContentValues();
        values.put(Events.DTSTART, freeTime);
        values.put(Events.DTEND, freeTime + 300000);
        values.put(Events.TITLE, "Запланированный звонок");
        values.put(Events.DESCRPTION, descr);
        values.put(Events.CALENDAR_ID, getCalendarID(context));
        values.put(Events.EVENT_TIMEZONE, TimeZone.getDefault().getID());
        values.put(Events.GUESTS_CAN_INVITE_OTHERS, "1");
        values.put(Events.GUESTS_CAN_SEE_GUESTS, "1");
        values.put(Events.ORGANIZER, "168irishka10@gmail.com");
        if (ActivityCompat.checkSelfPermission(context,
Manifest.permission.WRITE_CALENDAR) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        Uri eventUri = cr.insert(Events.CONTENT_URI, values);
        Log.i("LOOK HERE: CalendarHandler", descr);
        Log.i("LOOK HERE: CalendarHandler", "INSERTED Event ID: " +
eventUri.getLastPathSegment());
    }
}

public static void deleteEvent(Context context, String phoneNumber, long
callStartTime, long freeTime) {
    Uri eventUri = getEventUri(context, phoneNumber, callStartTime,
freeTime);
    if (eventUri != null) {
        ContentResolver cr = context.getContentResolver();
        cr.delete(eventUri, null, null);
        Log.i("LOOK HERE: CalendarHandler", "DELETED Event Uri: " +
eventUri);
    }
}

public static Uri getEventUri(Context context, String phoneNumber, long
callStartTime, long freeTime) {
    String descr =
context.getResources().getString(R.string.textCalendar, phoneNumber,
getTimeStringFromLong(callStartTime, "dd.MM.y HH:mm"));
    ContentResolver cr = context.getContentResolver();
    Uri uri = Events.CONTENT_URI;

    String[] from = {
        Events._ID,
        Events.TITLE,
        Events.DESCRPTION,
    };
    String where = "(" + Events.DESCRPTION + " = ?)";
    String[] equal = new String[] { descr };

    long eventID = -1;
    Cursor cur = cr.query(uri, from, where, equal, null);
    while (cur.moveToNext()) {
        eventID = cur.getLong(cur.getColumnIndex(Events._ID));
        Log.i("LOOK HERE: CalendarHandler", "CHECKED Event title: " +
cur.getString(cur.getColumnIndex(Events.TITLE)));
        Log.i("LOOK HERE: CalendarHandler", "CHECKED Event ID: " +

```

```

eventID);
    }
    cur.close();
    if (eventID == -1) {
        Log.i("LOOK HERE: CalendarHandler", "CHECK: new");
        return null;
    } else {
        Log.i("LOOK HERE: CalendarHandler", "CHECK: old");
        return ContentUris.withAppendedId(Events.CONTENT_URI, eventID);
    }
}

    public static String getTimeStringFromDate(Date SomeTimeDate, String
formatStyle) {
        SimpleDateFormat dateFormat = new SimpleDateFormat(formatStyle);
        //Маска даты
        //dateFormat.setTimeZone(TimeZone.getDefault());
        return dateFormat.format(SomeTimeDate);
    }

    public static String getTimeStringFromLong(long millis, String
formatStyle) {
        return getTimeStringFromDate(new Date(millis), formatStyle);
    }
}

```

DrawerActivity

```

package com.shilina.project_x;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.Html;
import android.util.Log;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

public class DrawerActivity extends AppCompatActivity {

    String className;
    DrawerLayout drawerLayout;
    TextView nicknameView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //Открыть новое окно
        public void redirectActivity(Activity activity, Class aclass){
            Intent intent = new Intent(activity, aclass);
            activity.startActivity(intent);
        }

        //Открыть боковое меню
        public void openDrawer(DrawerLayout drawerLayout) {
            if(!drawerLayout.isDrawerOpen(GravityCompat.START)) {

```

```

        drawerLayout.openDrawer(GravityCompat.START);
        Log.i("LOOK HERE: DA", "Side menu was OPENED");
    }
}
//Закреть боковое меню
public void closeDrawer(DrawerLayout drawerLayout) {
    if (drawerLayout.isDrawerOpen(GravityCompat.START)) {
        drawerLayout.closeDrawer(GravityCompat.START);
        Log.i("LOOK HERE: DA", "Side menu was CLOSED");
    }
}

public void onMenuClick(View view) {
    openDrawer(drawerLayout);
}

public void onLogoClick(View view) {
    closeDrawer(drawerLayout);
}

public void setNickname() {
    if (nicknameView == null) {
        nicknameView = findViewById(R.id.nickname);
        nicknameView.setText(SettingsActivity.getUser(this));
    }
}

public void onLaterActClick(View view) {
    if (className.equals(LaterCallsActivity.className)) {
        closeDrawer(drawerLayout);
        Log.i("LOOK HERE: DA", "Activity already opened");
    } else {
        redirectActivity(this, LaterCallsActivity.class);
        Log.i("LOOK HERE: DA", "Redirecting to LaterCalls");
    }
}

public void onStatusActClick(View view) {
    if (className.equals(StatusActivity.className)) {
        closeDrawer(drawerLayout);
        Log.i("LOOK HERE: DA", "Activity already opened");
    } else {
        redirectActivity(this, StatusActivity.class);
        Log.i("LOOK HERE: DA", "Redirecting to Status");
    }
}

public void onSettingsActClick(View view) {
    if (className.equals(SettingsActivity.className)) {
        closeDrawer(drawerLayout);
        Log.i("LOOK HERE: DA", "Activity already opened");
    } else {
        redirectActivity(this, SettingsActivity.class);
        Log.i("LOOK HERE: DA", "Redirecting to Settings");
    }
}

public void onHelpActClick(View view) {
    if (className.equals(HelpActivity.className)) {
        closeDrawer(drawerLayout);
        Log.i("LOOK HERE: DA", "Activity already opened");
    } else {
        redirectActivity(this, HelpActivity.class);
        Log.i("LOOK HERE: DA", "Redirecting to Help");
    }
}

```

```

    }
}

public void onAboutUsActClick(View view) {
    if (className.equals(AboutUsActivity.className)) {
        closeDrawer(drawerLayout);
        Log.i("LOOK HERE: DA", "Activity already opened");
    } else {
        redirectActivity(this, AboutUsActivity.class);
        Log.i("LOOK HERE: DA", "Redirecting to AboutUs");
    }
}

public void onLogoutClick(View view) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(Html.fromHtml("<font
color='#000000'>Logout</font>"));
    builder.setMessage(Html.fromHtml("<font color='#000000'>Are u
sure?</font>"));
    //builder.setTitle("Logout");
    //builder.setMessage("Are u sure?");

    builder.setPositiveButton(Html.fromHtml("<font
color='#000000'>YES</font>"), new DialogInterface.OnClickListener() {
        //builder.setPositiveButton("YES", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                SettingsActivity.setUser(getApplicationContext(), null);
                CalendarHandler.deleteCalendar(getApplicationContext());
                Intent service = new Intent(getApplicationContext(),
BroadcastService.class);
                stopService(service);
                finishAffinity();
            }
        });

    builder.setNegativeButton(Html.fromHtml("<font
color='#000000'>NO</font>"), new DialogInterface.OnClickListener() {
        //builder.setNegativeButton("NO", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });

    builder.show();
}

@Override
public void onBackPressed() {
    if(drawerLayout.isDrawerOpen(GravityCompat.START)) {
        drawerLayout.closeDrawer(GravityCompat.START);
        Log.i("LOOK HERE: DA", "Side menu was CLOSED");
    } else {
        Log.i("LOOK HERE: DA", "App was closed");
        finishAffinity();
    }
}
}
}

```

HelpActivity

```

package com.shilina.project_x;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;

public class HelpActivity extends DrawerActivity {

    public static final String className =
Thread.currentThread().getStackTrace()[2].getClassName();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SettingsActivity.chooseTheme(this);
        setContentView(R.layout.activity_help);
        super.drawerLayout = findViewById(R.id.drawer_layout);
        super.className = className;
        setNickname();
        Log.i("LOOK HERE: HA", "Opened activity is: " + className);

        TextView textById = findViewById(R.id.nameOfWindow);
        textById.setText("Помощь");
    }
}

```

LaterCallsActivity

```

package com.shilina.project_x;

import androidx.drawerlayout.widget.DrawerLayout;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONException;
import org.json.JSONObject;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;

import static android.util.TypedValue.COMPLEX_UNIT_SP;

public class LaterCallsActivity extends DrawerActivity {

    public static final String className =

```



```

Thread.currentThread().getStackTrace()[2].getClassName();
    ViewGroup oneCallLayout;
    public static boolean isAddView = false;
    public static PlanCallLayout planLayout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SettingsActivity.chooseTheme(this);
        setContentView(R.layout.activity_later_calls);
        super.drawerLayout = findViewById(R.id.drawer_layout);
        super.className = className;
        setNickname();
        Log.i("LOOK HERE: LCA", "Opened activity is: " + className);

        //Assign variable
        TextView textById = findViewById(R.id.nameOfWindow);
        textById.setText("Запланированные звонки");

        oneCallLayout = (ViewGroup) findViewById(R.id.scroll_view_layout);

        //Кнопка "Запланировать звонок"
        View planOneCall =
        getLayoutInflater().inflate(R.layout.one_call_view, oneCallLayout, false);
        planOneCall.setId(0);
        planOneCall.setTag("plan");
        TextView plannedCallViewText = (TextView)
        planOneCall.findViewById(R.id.one_call_view_text);
        plannedCallViewText.setText("Чтобы запланировать звонок, нажмите:");
        Button plannedCallViewButton = (Button)
        planOneCall.findViewById(R.id.one_call_view_button);
        plannedCallViewButton.setText("Запланировать");
        plannedCallViewButton.setTextSize(COMPLEX_UNIT_SP, 8);
        plannedCallViewButton.setOnClickListener(new Button.OnClickListener()
        {
            @Override
            public void onClick(View view) {
                if (planLayout == null) {
                    planLayout = new PlanCallLayout(getApplicationContext(),
                    LaterCallsActivity.this, "", null);
                }
                if (!planLayout.isShown) {
                    planLayout.addBubble();
                    isAddView = false;
                    //TODO: Назначить закрытие окна кнопкой назад
                }
            }
        });
        oneCallLayout.addView(planOneCall);
        Log.i("LOOK HERE: LCA", "poc number is: " + planOneCall.getId());

        //Вывод всех View в Layout'e
        /*
        for (int j = 0; j < oneCallLayout.getChildCount(); j++) {
            Log.i("LOOK HERE: LCA", "Child: " + oneCallLayout.getChildAt(j));
        }
        */
    }

    @Override
    protected void onResume() {
        super.onResume();
        oneCallLayout.removeViews(1, oneCallLayout.getChildCount()-1);
    }

```

```

//Получение звонков
Runnable backgroundProcess = new Runnable() {
    public void run() {
        try {
            HashMap<String, String> data = new HashMap<String,
String>() {{
                put("email",
SettingsActivity.getUser(getApplicationContext()));
            }};
            ServerHandler loginQuery = new
ServerHandler(ServerHandler.ACTION_GET_RESCHEDULING, data);
            loginQuery.execute();
            String responseString = loginQuery.get();
            JSONObject responseJSON = new JSONObject(responseString);
            Context context = getApplicationContext();
            if (ServerHandler.isErrored(responseString) == null) {
                JSONObject plannedCallsArray =
responseJSON.getJSONObject("calls");
                for (int i = 1; i <= plannedCallsArray.length(); i++)
                {
                    try {
                        JSONObject calli =
plannedCallsArray.getJSONObject(Integer.toString(i));
                        int id =
Integer.parseInt(calli.getString("id"));
                        String phone = calli.getString("phone");
                        Date callDate = new
Date(Long.valueOf(calli.getString("call_datetime").concat("000")));
                        Date callbackDate = new
Date(Long.valueOf(calli.getString("callback_datetime").concat("000")));

                        CalendarHandler.addEvent(context, phone,
callDate.getTime(), callbackDate.getTime());

                        View nextOneCall =
getLayoutInflater().inflate(R.layout.one_call_view, oneCallLayout, false);
                        nextOneCall.setId(i);
                        nextOneCall.setTag("" + id + ":" +
callbackDate.getTime());
                        TextView oneCallViewText = (TextView)
nextOneCall.findViewById(R.id.one_call_view_text);

                        oneCallViewText.setText(context.getResources().getString(R.string.textOneCall
, phone, CalendarHandler.getTimeStringFromDate(callDate, "dd.MM.y
HH:mm"), CalendarHandler.getTimeStringFromDate(callbackDate, "dd.MM.y
HH:mm")));

                        Button oneCallViewButton = (Button)
nextOneCall.findViewById(R.id.one_call_view_button);
                        oneCallViewButton.setOnClickListener(new
Button.OnClickListener() {
                            @Override
                            public void onClick(View view) {
                                View removedOneCall = (View)
view.getParent();

                                oneCallLayout.removeView(removedOneCall);
                                Log.i("LOOK HERE: LCA", "Removing poc
number: " + removedOneCall.getId());

                                Runnable backgroundProcess = new

Runnable() {
                                    public void run() {
                                        try {
                                            HashMap<String, String>

```

```

data = new HashMap<String, String>() {{
    put("id_call",
nextOneCall.getTag().toString().split(":")[0]);
    }};
    ServerHandler
addCallQuery = new ServerHandler(ServerHandler.ACTION_DELETE_CALL, data);
    addCallQuery.execute();
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
    };
    Thread thread = new Thread(null,
backgroundProcess, "Background");
    thread.start();
    CalendarHandler.deleteEvent(context,
phone, callDate.getTime(), callbackDate.getTime());
    String message =
context.getResources().getString(R.string.cancelSMS, callDate);
    SMSHandler.sendSMS(context, phone,
message);
    }
    });
    new Handler(Looper.getMainLooper()).post(new
Runnable() {
        public void run() {
            oneCallLayout.addView(nextOneCall);
        }
    });
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
    } else {
        new Handler(Looper.getMainLooper()).post(new
Runnable() {
            public void run() {
                try {
                    Toast.makeText(context, "Данные не
загружены: " + responseJSON.getString("error_text"),
Toast.LENGTH_SHORT).show();
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
    };
    Thread thread = new Thread(null, backgroundProcess, "Background");
    thread.start();
}

@Override
protected void onPause() {
    super.onPause();
    closePCL();
}

@Override

```

```

    public void openDrawer(DrawerLayout drawerLayout) {
        super.openDrawer(drawerLayout);
        closePCL();
    }

    static void closePCL() {
        if (planLayout != null) {
            planLayout.removePCL();
            planLayout = null;
        }
    }
}

```

LoginActivity

```

package com.shilina.project_x;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.File;
import java.util.HashMap;
import java.util.Map;

import static java.lang.String.valueOf;

public class LoginActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (SettingsActivity.isAuthorised(this)) {
            Intent intent = new Intent(this, LaterCallsActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
        } else {
            SettingsActivity.chooseTheme(this);
            setContentView(R.layout.activity_login);
        }
    }

    public void CheckLogin(View view) {
        String username = ((EditText)
        findViewById(R.id.login)).getText().toString();
        EditText et_password = (EditText) findViewById(R.id.password);
        et_password.clearFocus();
        String android_id = Settings.Secure.getString(getContentResolver(),
        Settings.Secure.ANDROID_ID);

        //Обращение к серверу
        Runnable backgroundProcess = new Runnable() {
            public void run() {

```

```

        try {
            HashMap<String, String> data = new HashMap<String,
String>() {{
                put("email", username);
                put("pass", et_password.getText().toString());
                put("android_id", android_id);
            }};
            ServerHandler loginQuery = new
ServerHandler(ServerHandler.ACTION_LOGIN, data);
            loginQuery.execute();
            String responseString = loginQuery.get();
            JSONObject responseJSON = new JSONObject(responseString);
            Context context = getApplicationContext();
            if (ServerHandler.isErrored(responseString) == null) {
                SettingsActivity.setUser(context, data.get("email"));
                SettingsActivity.setPremium(context,
Boolean.parseBoolean(responseJSON.getString("user_premium")));
                new Handler(Looper.getMainLooper()).post(new
Runnable() {
                    public void run() {
                        Toast.makeText(context, "Вход выполнен",
Toast.LENGTH_SHORT).show();
                    }
                });
                Intent intent = new Intent(context,
LaterCallsActivity.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                context.startActivity(intent);
            } else {
                new Handler(Looper.getMainLooper()).post(new
Runnable() {
                    public void run() {
                        try {
                            Toast.makeText(context, "Ошибка
авторизации: " + responseJSON.getString("error_text"),
Toast.LENGTH_SHORT).show();
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                });
            }
        } catch (Exception e) {
        }
    }

    Thread thread = new Thread(null, backgroundProcess, "Background");
    thread.start();

    /*
    if (username.equals(user) &&
et_password.getText().toString().equals(hash)) {
        SettingsActivity.setUser(this, username);
        SettingsActivity.setPremium(this, status);
        Toast.makeText(getApplicationContext(), "Вход выполнен!",
Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(this, LaterCallsActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        finish();
    }
    */
}

```

```

    public void GoSignUp(View view) {
        Intent intentToSignUp = new Intent(this, SignUpActivity.class);
        startActivity(intentToSignUp);
    }

    @Override
    public void onBackPressed() {
        Log.i("LOOK HERE: DA", "App was closed");
        finishAffinity();
    }
}

```

OneCall

```

package com.shilina.project_x;

import android.net.Uri;
import android.util.Log;

import java.net.URI;
import java.util.Date;

public class OneCall {

    int id;
    String caller;
    Date callStartTime;
    Date callPlannedTime;

    public OneCall(int id, String caller, Date callStartTime, Date
callPlannedTime) {
        this.id = id;
        this.caller = caller;
        this.callStartTime = callStartTime;
        this.callPlannedTime = callPlannedTime;
        Log.i("LOOK HERE: OC", "callID: " + id + "\nCall with: " + caller +
"\nWas planned: " + callStartTime + "\nWill: " + callPlannedTime);
    }
}

```

PermissionsActivity

```

package com.shilina.project_x;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.view.GravityCompat;

import android.Manifest;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.PowerManager;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.Switch;
import android.widget.Toast;

```

```

import java.util.ArrayList;

public class PermissionsActivity extends AppCompatActivity {

    Switch permSwitcher;
    public String callingActivity;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SettingsActivity.chooseTheme(this);
        setContentView(R.layout.activity_permissions);
        permSwitcher = findViewById(R.id.switch1);
        //Метод обработки смены значения кнопки
        permSwitcher.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                if (permSwitcher.isChecked()) {
                    if (!isPermissionsGranted(all_permissions_str)) {
                        givePermissions();
                    }
                }
            }
        });

        callingActivity = getIntent().getStringExtra("calling-activity");
        Log.i("LOOK HERE: PermissionsActivity", "CALLING ACT IS: " +
callingActivity);
        Log.i("LOOK HERE: PermissionsActivity", "Settings className IS: " +
SettingsActivity.className);

        if (isPermissionsGranted(all_permissions_str)) {
            if (android.provider.Settings.canDrawOverlays(this)) {
                if (callingActivity == null) {
                    onContinueClick(new View(getApplicationContext()));
                }
            }
        } else {
            permSwitcher.setChecked(false);
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        if (isPermissionsGranted(all_permissions_str)) {
            permSwitcher.setChecked(true);
        } else {
            permSwitcher.setChecked(false);
            Intent service = new Intent(getApplicationContext(),
BroadcastService.class);
            stopService(service);
            Toast.makeText(getApplicationContext(), "Еще не ВСЕ разрешения
предоставлены", Toast.LENGTH_SHORT).show();
        }
    }

    //Метод, блокирующий выключение кнопки
    public void onPermClick(View view) {
        if (!permSwitcher.isChecked()) {
            Toast.makeText(getApplicationContext(), "Отозвать разрешения

```

```

        можно только в настройках устройства", Toast.LENGTH_SHORT).show();
        permSwitcher.setChecked(true);
    }

    public void onContinueClick(View view) {
        if (isPermissionsGranted(all_permissions_str) &&
            android.provider.Settings.canDrawOverlays(this)) {
            Intent service = new Intent(this, BroadcastService.class);
            this.startService(service);
            Intent intent = new Intent(this, LoginActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
            Toast.makeText(getApplicationContext(), "Разрешения находятся в
разделе \"Настройки\"", Toast.LENGTH_SHORT).show();
        } else {
            Intent service = new Intent(getApplicationContext(),
BroadcastService.class);
            stopService(service);
            Toast.makeText(getApplicationContext(), "Еще не ВСЕ разрешения
предоставлены", Toast.LENGTH_SHORT).show();
        }
    }

    //Списки с разрешениями
    ArrayList<String> needed_permissions_list = new ArrayList<>();
    String[] all_permissions_str = {Manifest.permission.READ_CONTACTS,
Manifest.permission.READ_PHONE_STATE,
Manifest.permission.ANSWER_PHONE_CALLS,
Manifest.permission.READ_CALL_LOG,
Manifest.permission.SEND_SMS,
Manifest.permission.INTERNET,
Manifest.permission.ACCESS_NETWORK_STATE,
Manifest.permission.READ_CALENDAR,
Manifest.permission.WRITE_CALENDAR,
Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE,
};

    //Метод, проверяющий базовые разрешения
    public boolean isPermissionsGranted(String[] all_permissions_str) {
        needed_permissions_list = new ArrayList<> ();
        for (String permission : all_permissions_str) {
            int permission_status = ContextCompat.checkSelfPermission(this,
permission); //Проверяем текущее разрешение
            if (permission_status != PackageManager.PERMISSION_GRANTED) {
                //Если разрешение не получено, =>
                Log.i("LOOK HERE: PermissionsActivity", permission+" is
NOT");
                needed_permissions_list.add(permission); //то добавляем его в
список неполученных
            } else {
                Log.i("LOOK HERE: PermissionsActivity", permission+" is
GRANTED");
            }
        }
        if (needed_permissions_list.size() == 0) {
            return true;
        } else {
            return false;
        }
    }

    //Метод, запрашивающий базовые разрешения

```



```

    public void givePermissions() {
        String[] needed_permission_str = needed_permissions_list.toArray(new
String[0]);
        ActivityCompat.requestPermissions(this, needed_permission_str, 1);
        for (String perm : needed_permission_str) {
            Log.i("LOOK HERE: PermissionsActivity", perm + " is NOW");
        }
    }

    //Метод, запрашивающий разрешение на показ поверх остальных приложений
    public void onOverClick(View view) {
        if (!android.provider.Settings.canDrawOverlays(this)) {
            Intent intent = new
Intent(android.provider.Settings.ACTION_MANAGE_OVERLAY_PERMISSION,
Uri.parse("package:" + getPackageName()));
            startActivity(intent);
            Log.i("LOOK HERE: PermissionsActivity", "Overlay is NOW");
        } else {
            Toast.makeText(this, "Разрешение уже предоставлено",
Toast.LENGTH_SHORT).show();
        }
    }

    //Метод, запрашивающий разрешение на работу в фоновом режиме
    public void onBackClick(View view) {
        PowerManager powerMan = (PowerManager)
getSystemService(Context.POWER_SERVICE);
        if (!powerMan.isIgnoringBatteryOptimizations(getPackageName())) {
            Log.i("LOOK HERE: PermissionsActivity", "Battery Saver is " +
powerMan.isIgnoringBatteryOptimizations(getPackageName()));
            Intent intent = new
Intent(android.provider.Settings.ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS,
Uri.parse("package:" + getPackageName()));
            if (getPackageManager().resolveActivity(intent,
PackageManager.MATCH_DEFAULT_ONLY) != null) {
                startActivity(intent);
            } else {
                Toast.makeText(this, "На вашем телефоне это не нужно",
Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(this, "Разрешение уже предоставлено",
Toast.LENGTH_SHORT).show();
        }
    }

    //Метод, запрашивающий отключение оптимизации энергопотребления
    public void onBatClick(View view) {
        Intent intent = new Intent();
        intent.setComponent(new ComponentName("com.miui.powerkeeper",
"com.miui.powerkeeper.ui.HiddenAppsConfigActivity"));
        intent.putExtra("package_name", getPackageName());
        intent.putExtra("package_label", getText(R.string.app_name));
        if (getPackageManager().resolveActivity(intent,
PackageManager.MATCH_DEFAULT_ONLY) != null) {
            startActivity(intent);
            Toast.makeText(this, "Пожалуйста, выберите \"Нет ограничений\"",
Toast.LENGTH_LONG).show();
        } else {
            intent = new Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS,
Uri.parse("package:" + getPackageName()));
            if (getPackageManager().resolveActivity(intent,
PackageManager.MATCH_DEFAULT_ONLY) != null) {
                startActivity(intent);
            }
        }
    }

```

```

        Toast.makeText(this, "Пожалуйста, \notключите оптимизацию
энергопотребления\ндля приложения Project_X", Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(this, "На вашем телефоне это не нужно",
Toast.LENGTH_SHORT).show();
    }
}
Log.i("LOOK HERE: PermissionsActivity", "Battery Saver is NOW");
}

//Метод, запрашивающий отключение оптимизации энергопотребления
public void onAutoClick(View view) {
    Intent[] POWERMANAGER_INTENTS = {
        new Intent().setComponent(new
ComponentName("com.miui.securitycenter",
"com.miui.permcenter.autostart.AutoStartManagementActivity")),
        new Intent().setComponent(new
ComponentName("com.letv.android.letvsafe",
"com.letv.android.letvsafe.AutobootManageActivity")),
        new Intent().setComponent(new
ComponentName("com.huawei.systemmanager",
"com.huawei.systemmanager.optimize.process.ProtectActivity")),
        new Intent().setComponent(new
ComponentName("com.huawei.systemmanager",
"com.huawei.systemmanager.appcontrol.activity.StartupAppControlActivity")),
        new Intent().setComponent(new
ComponentName("com.huawei.systemmanager",
"com.huawei.systemmanager.startupmgr.ui.StartupNormalAppListActivity")),
        new Intent().setComponent(new
ComponentName("com.coloros safecenter",
"com.coloros safecenter.permission.startup.StartupAppListActivity")),
        new Intent().setComponent(new
ComponentName("com.coloros safecenter",
"com.coloros safecenter.startupapp.StartupAppListActivity")),
        new Intent().setComponent(new ComponentName("com.oppo.safe",
"com.oppo.safe.permission.startup.StartupAppListActivity")),
        new Intent().setComponent(new
ComponentName("com.iqoo.secure",
"com.iqoo.secure.ui.phoneoptimize.AddWhiteListActivity")),
        new Intent().setComponent(new
ComponentName("com.iqoo.secure",
"com.iqoo.secure.ui.phoneoptimize.BgStartUpManager")),
        new Intent().setComponent(new
ComponentName("com.vivo.permissionmanager",
"com.vivo.permissionmanager.activity.BgStartUpManagerActivity")),
        new Intent().setComponent(new
ComponentName("com.samsung.android.lool",
"com.samsung.android.sm.ui.battery.BatteryActivity")),
        new Intent().setComponent(new
ComponentName("com.htc.pitroad",
"com.htc.pitroad.landingpage.activity.LandingPageActivity")),
        new Intent().setComponent(new
ComponentName("com.asus.mobilemanager",
"com.asus.mobilemanager.MainActivity"))};

    boolean powers = true;
    for (Intent intent : POWERMANAGER_INTENTS) {
        if (getPackageManager().resolveActivity(intent,
PackageManager.MATCH_DEFAULT_ONLY) != null) {
            startActivity(intent);
            Toast.makeText(this, "Пожалуйста, включите Автозапуск для
приложения Project_X", Toast.LENGTH_LONG).show();
            powers = false;
            break;

```

```

    }
    }
    if (powers) {
        Toast.makeText(this, "На вашем телефоне это не нужно",
Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onBackPressed() {
    Log.i("LOOK HERE: DA", "App was closed");
    finishAffinity();
}
}

```

PhoneHandler

```

package com.shilina.project_x;

import android.Manifest;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.PixelFormat;
import android.os.Build;
import android.telecom.TelecomManager;
import android.telephony.TelephonyManager;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.WindowManager;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.RequiresApi;
import androidx.core.app.ActivityCompat;

import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;

public class PhoneHandler extends BroadcastReceiver {

    //Счет экземпляров
    public static int index = 0;
    public static boolean isRegistered = false;

    //Переменные состояния
    public static int statePre = TelephonyManager.CALL_STATE_IDLE;
    public static int stateCur = TelephonyManager.CALL_STATE_IDLE;

    //Переменные отдельного звонка
    public static String phoneNumber;
    public static Date callStartTime;
    public static boolean isIncoming = false;
    public static PlanCallLayout callLayout;

    @Override
    public void onReceive(Context context, Intent intent) {
        Log.i("LOOK HERE: PhoneHandler", index++ + " example of PH has been

```

```

created");

        //Определение нового состояния телефона относительно звонка
        String state_str =
intent.getExtras().getString(TelephonyManager.EXTRA_STATE);
        int stateNew = TelephonyManager.CALL_STATE_IDLE;
        if (state_str.equals(TelephonyManager.EXTRA_STATE_IDLE)) {
            stateNew = TelephonyManager.CALL_STATE_IDLE;
        } else if (state_str.equals(TelephonyManager.EXTRA_STATE_RINGING)) {
            stateNew = TelephonyManager.CALL_STATE_RINGING;
        } else if (state_str.equals(TelephonyManager.EXTRA_STATE_OFFHOOK)) {
            stateNew = TelephonyManager.CALL_STATE_OFFHOOK;
        }
        onCallStateChanged(context, intent, stateNew);
    }

    //Метод обработки изменения текущего состояния телефона относительно
звонка
    //Входящие вызовы: при звонке RINGING (1), при ответе OFFHOOK (2), при
отклонении/завершении IDLE (0)
    //Исходящие вызовах: при наборе OFFHOOK (2), при отклонении/завершении
IDLE (0)
    public void onCallStateChanged(Context context, Intent intent, int
stateNew) {
        Log.i("LOOK HERE: PhoneHandler", "Previous state is: " + statePre +
", Current state is: " + stateCur + ", New state is: " + stateNew);
        if (stateCur == stateNew && stateCur != statePre) {
            switch (stateNew) {
                case TelephonyManager.CALL_STATE_RINGING:
                    //0 > 1 начало входящего вызова
                    phoneNumber =
intent.getStringExtra(TelephonyManager.EXTRA_INCOMING_NUMBER);
                    callStartTime = Calendar.getInstance().getTime();
                    isIncoming = true;
                    onIncomingCallReceived(context, phoneNumber,
callStartTime);
                    break;
                case TelephonyManager.CALL_STATE_OFFHOOK:
                    //1 > 2 начало входящего разговора
                    if (isIncoming) {
                        onIncomingCallAnswered(context, phoneNumber,
callStartTime);
                    }
                    //0 > 2 начало исходящего вызова
                    } else {
                        phoneNumber =
intent.getExtras().getString("android.intent.extra.PHONE_NUMBER");//НЕ
РАБОТАЕТ!!!
                        callStartTime = new Date();
                        onOutgoingCallStarted(context, phoneNumber,
callStartTime);
                    }
                    break;
                case TelephonyManager.CALL_STATE_IDLE:
                    // 1 > 0 отклонение входящего вызова
                    if (statePre == TelephonyManager.CALL_STATE_RINGING) {
                        onMissedCall(context, phoneNumber, callStartTime);
                        // 2 > 0 окончание входящего разговора
                    } else if (isIncoming) {
                        onIncomingCallEnded(context, phoneNumber,
callStartTime, Calendar.getInstance().getTime());
                        // 2 > 0 окончание исходящего разговора/вызова
                    } else {
                        onOutgoingCallEnded(context, phoneNumber,
callStartTime, Calendar.getInstance().getTime());
                    }
                }
            }
        }
    }

```

```

        }
        isIncoming = false;
        break;
    }
    Log.i("LOOK HERE: PhoneHandler", "The phone number is: " +
phoneNumber);
    } else {
        Log.i("LOOK HERE: PhoneHandler", "This is mid-state");
    }
    statePre = stateCur;
    stateCur = stateNew;
}

//Обработка начала входящего звонка
public void onIncomingCallReceived(Context context, String phoneNumber,
Date startTime) {
    Log.i("LOOK HERE: PhoneHandler", "Incoming call has been received");
    if (SettingsActivity.isAuthorised(context)) {
        SharedPreferences sp_settings =
context.getSharedPreferences(SettingsActivity.SP_FILE, Context.MODE_PRIVATE);
        String curModeIn =
sp_settings.getString(SettingsActivity.SP_MODE_IN,
SettingsActivity.SP_MODES_IN[0]);
        String curModeOut =
sp_settings.getString(SettingsActivity.SP_MODE_OUT,
SettingsActivity.SP_MODES_OUT[0]);
        boolean isBusyNow = !CalendarHandler.isFreeAt(context,
startTime.getTime());

        if (!isBusyNow &&
(curModeOut.equals(SettingsActivity.SP_MODES_OUT[1]))) {
            Log.i("LOOK HERE: PhoneHandler", "Mode for Out = OFF");
        } else if (isBusyNow &&
(curModeIn.equals(SettingsActivity.SP_MODES_IN[1]))) {
            Log.i("LOOK HERE: PhoneHandler", "Mode for In = Auto");
            long timeToSetMillis =
CalendarHandler.getFreeTimeFromCalendar(context, startTime.getTime());

            Runnable backgroundProcess = new Runnable() {
                public void run() {
                    try {
                        HashMap<String, String> data = new
HashMap<String, String>() {{
                            put("email",
SettingsActivity.getUser(context));
                            put("recipient_number", phoneNumber);
                            put("call_datetime",
Long.toString(startTime.getTime() / 1000));
                            put("callback_datetime",
Long.toString(timeToSetMillis / 1000));
                        }};
                        ServerHandler addCallQuery = new
ServerHandler(ServerHandler.ACTION_SET_RESCHEDULING, data);
                        addCallQuery.execute();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            };
            Thread thread = new Thread(null,
backgroundProcess, "Background");
            thread.start();
            CalendarHandler.addEvent(context, phoneNumber,
startTime.getTime(), timeToSetMillis);

```

```

        String message =
context.getResources().getString(R.string.textSMS,
CalendarHandler.getTimeStringFromLong(timeToSetMillis, "dd.MM.y HH:mm"));
        SMSHandler.sendSMS(context, phoneNumber, message);
        endRingingCall(context,phoneNumber);
    } else {
        Log.i("LOOK HERE: PhoneHandler", "Mode = Hand");
        callLayout = new PlanCallLayout(context, PhoneHandler.this,
phoneNumber, Calendar.getInstance().getTime());
        callLayout.addButton();
    }
    } else {
        Toast.makeText(context, "Пожалуйста, авторизуйтесь в Project_X",
Toast.LENGTH_SHORT).show();
    }
}

//Обработка начала входящего разговора
public void onIncomingCallAnswered(Context context, String phoneNumber,
Date startTime) {
    Log.i("LOOK HERE: PhoneHandler", "Incoming call has been answered");
    if (callLayout != null) {
        callLayout.removePCL(); //Удаляем всплывающее окно
    }
}

//Обработка окончания входящего разговора
public void onIncomingCallEnded(Context context, String phoneNumber, Date
startTime, Date endTime) {
    Log.i("LOOK HERE: PhoneHandler", "Incoming call has been ended");
    clearAll();
}

//Обработка начала исходящего звонка/разговора
public void onOutgoingCallStarted(Context context, String phoneNumber,
Date startTime) {
    Log.i("LOOK HERE: PhoneHandler", "Outgoing call has been
started/answered");
}

//Обработка окончания исходящего разговора
public void onOutgoingCallEnded(Context context, String phoneNumber, Date
startTime, Date endTime) {
    Log.i("LOOK HERE: PhoneHandler", "Outgoing call has been ended");
}

//Обработка окончания входящего вызова (сброс)
public void onMissedCall(Context context, String phoneNumber, Date
startTime) {
    Log.i("LOOK HERE: PhoneHandler", "Incoming call has been declined or
missed");
    clearAll();
}

//Очистка сессии
public void clearAll() {
    //Восстанавливаем переменные
    index = 0;
    phoneNumber = null;
    callStartTime = null;
    if (callLayout != null) {
        callLayout.removePCL();
        callLayout = null;
    }
}

```

```

    }

    public static void endRingCall(Context context, String phone){
        if (ActivityCompat.checkSelfPermission(context,
Manifest.permission.ANSWER_PHONE_CALLS) == PackageManager.PERMISSION_GRANTED)
        {
            if (phone.equals(phoneNumber) && stateCur ==
TelephonyManager.CALL_STATE_RINGING) {
                TelecomManager teleMan = (TelecomManager)
context.getSystemService(Context.TELECOM_SERVICE);
                teleMan.endCall();
                Log.i("LOOK HERE: PCL", "Звонок завершен");
            }
        }
    }
}

```

PlanCallLayout

```

package com.shilina.project_x;

import android.app.Activity;
import android.app.Application;
import android.content.Context;
import android.graphics.Insets;
import android.graphics.PixelFormat;
import android.icu.util.GregorianCalendar;
import android.os.Build;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.WindowInsets;
import android.view.WindowManager;
import android.view.WindowMetrics;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;

import android.widget.Toast;
import android.widget.TimePicker;

import java.util.regex.Pattern;

import androidx.annotation.RequiresApi;

public class PlanCallLayout {

    public WindowManager windowMan;
    public LayoutInflater layInflater;
    public WindowManager.LayoutParams layPar;
    public ViewGroup vgLayout;
    public int width;
    public int height;
    public Context context;
    public boolean isShown;
}

```

```

Object parentObject;

public String phoneNumber;
public Date callStartTime;

//Создание управления окном
public PlanCallLayout(Context context, Object parentObject, String
phoneNumber, Date callStartTime) {
    Log.i("LOOK HERE: PCL", "Inflater has been created");
    this.context = context;
    this.parentObject = parentObject;
    this.phoneNumber = phoneNumber;
    if (callStartTime != null) {
        this.callStartTime = callStartTime;
    } else {
        this.callStartTime = Calendar.getInstance().getTime();
    }
    SettingsActivity.chooseTheme(context);
    windowMan = (WindowManager)
context.getSystemService(Context.WINDOW_SERVICE); //Получаем сервис
управления окном
    layInflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE); //Получаем сервис
управления макетом

    width = getActivitySize(context)[0]; //Ширина дисплея
    height = getActivitySize(context)[1]; //Высота дисплея
}

//Метод добавления кнопки
public void addButton() {
    Log.i("LOOK HERE: PCL", "Button has been created");
    //Присваиваем картинке обработчик нажатия
    vgLayout = (ViewGroup) layInflater.inflate(R.layout.call_logo_layout,
null);

    layPar = new WindowManager.LayoutParams( //Задаем параметры
отображения
        (int) (width / 4), (int) (height / 4),
        WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY, //
Говорим, что приложение будет поверх других
        WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE, // Необходимо
для того чтобы TouchEvent'ы в пустой области передавались на другие
приложения
        PixelFormat.TRANSLUCENT); // Само окно прозрачное

    windowMan.addView(vgLayout, layPar);
    isShown = true;

    ImageView button_img = vgLayout.findViewById(R.id.call_logo_button);
    button_img.setOnClickListener(new View.OnClickListener() {
        @RequiresApi(api = Build.VERSION_CODES.O_MR1)
        @Override
        public void onClick(View v) {
            Log.i("LOOK HERE: PCL", "Button has been touched!");
            removePCL();
            addBubble();
        }
    });
}

//Метод добавления текстового пузыря
public void addBubble() {
    Log.i("LOOK HERE: PCL", "Bubble has been created");

```



```

        vgLayout = (ViewGroup) layInflater.inflate(R.layout.plan_call_layout,
null);

        Log.i("LOOK HERE: PCL", "Ширина\n\tэкрана: " + width + "\n\tтокна: " +
width * 90 / 100);
        Log.i("LOOK HERE: PCL", "Высота\n\tэкрана: " + height + "\n\tтокна: "
+ height * 30 / 100);

        layPar = new WindowManager.LayoutParams( //Задаем параметры
отображения
            width, height,
            WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY, //
Говорим, что приложение будет поверх других
            WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL, //
Необходимо для того чтобы TouchEvent'ы в пустой области передавались на
другие приложения
            PixelFormat.TRANSLUCENT); // Само окно прозрачное

        windowMan.addView(vgLayout, layPar);
        isShown = true;

        DatePicker datePicker = (DatePicker)
vgLayout.findViewById(R.id.bubble_date_picker);
        TimePicker timePicker =
vgLayout.findViewById(R.id.bubble_time_picker);
        timePicker.setIs24HourView(true);
        LinearLayout lLayout = vgLayout.findViewById(R.id.bubble_left);
        TextView textToSend = (TextView)
vgLayout.findViewById(R.id.bubble_text);

        final long callStartTimeMillis = callStartTime.getTime();
        final long timeToSetMillis[] = new long[1];
        timeToSetMillis[0] = CalendarHandler.getFreeTimeFromCalendar(context,
callStartTimeMillis);

        TextView textPhone = (TextView)
vgLayout.findViewById(R.id.bubble_phone);
        textPhone.setText(phoneNumber);

        GregorianCalendar c = new GregorianCalendar();
        datePicker.setOnDateChangeListener(new
DatePicker.OnDateChangeListener() {
            @Override
            public void onDateChanged(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {
                c.set(year, monthOfYear, dayOfMonth, 0, 0, 0);
                timeToSetMillis[0] = c.getTimeInMillis();
            }
        });

        textToSend.setText(context.getResources().getString(R.string.textSMS,
CalendarHandler.getTimeStringFromLong(c.getTimeInMillis(), "dd.MM.y")));
    });

    boolean[] isConfirmed = new boolean[]{false};
    timePicker.setOnTimeChangeListener(new
TimePicker.OnTimeChangeListener() {
        @Override
        public void onTimeChanged(TimePicker view, int hours, int
minutes) {
            c.set(Calendar.HOUR_OF_DAY, hours);
            c.set(Calendar.MINUTE, minutes);
            timeToSetMillis[0] = c.getTimeInMillis();
        }
    });

    textToSend.setText(context.getResources().getString(R.string.textSMS,

```

```

CalendarHandler.getTimeStringFromLong(c.getTimeInMillis(), "dd.MM.y
HH:mm"));
        isConfirmed[0] = false;
    }
});

Button buttonSend = vgLayout.findViewById(R.id.bubble_button_send);
Button buttonReturn =
vgLayout.findViewById(R.id.bubble_button_return);
    if (timeToSetMillis[0] < 0) {
        lLayout.removeView(timePicker);

textToSend.setText(context.getResources().getString(R.string.textSMS,
"[Выберите дату и время]"));
    } else {
        lLayout.removeView(datePicker);

textToSend.setText(context.getResources().getString(R.string.textSMS,
CalendarHandler.getTimeStringFromLong(timeToSetMillis[0], "dd.MM.y HH:mm"));
        String[] hms =
CalendarHandler.getTimeStringFromLong(timeToSetMillis[0],
"HH:mm").split(":");
        timePicker.setHour(Integer.valueOf(hms[0]));
        timePicker.setMinute(Integer.valueOf(hms[1]));
        buttonSend.setText("Отправить");
        buttonReturn.setText("Назад");
    }

buttonSend.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String bst = buttonSend.getText().toString();
        if (bst.equals("Далее")) {
            if (timeToSetMillis[0] < 0) {
                Toast.makeText(context, "Введите верную дату",
Toast.LENGTH_SHORT).show();
            } else {
                lLayout.removeView(datePicker);
                lLayout.addView(timePicker, 1);
                buttonSend.setText("Отправить");
                buttonReturn.setText("Назад");
            }
            Log.i("LOOK HERE: PCL", "Далее");
        } else if (bst.equals("Отправить")) {
            Log.i("LOOK HERE: PCL", "Отправить");
            String numberToSend = textPhone.getText().toString();
            if (!Pattern.matches("^((8|\\+7)[\\-
]?)?(\\(?!\\d{3}\\)?)?([\\- ]?)?([\\d\\- ]{7,10})$", numberToSend) ||
(textToSend.getText().toString().split(":").length != 3)) {
                Log.i("LOOK HERE: PCL", "Номер или время введены
некорректно");
                Toast.makeText(context, "Введите корректный номер
телефона и время", Toast.LENGTH_SHORT).show();
            } else {
                if (!CalendarHandler.isFreeAt(context,
timeToSetMillis[0]) && !isConfirmed[0]) {
                    Toast.makeText(context, "На выбранное время уже
запланировано другое мероприятие.\nНажмите \"Отправить\", чтобы подтвердить",
Toast.LENGTH_SHORT).show();
                    isConfirmed[0] = true;
                } else {
                    Runnable backgroundProcess = new Runnable() {
                        public void run() {
                            try {

```

```

HashMap<String, String> data = new
HashMap<String, String>() {{
    put("email",
SettingsActivity.getUser(context));
    put("recipient_number",
numberToSend);
    put("call_datetime",
Long.toString(callStartTimeMillis / 1000));
    put("callback_datetime",
Long.toString(timeToSetMillis[0] / 1000));
    }};
ServerHandler addCallQuery = new
ServerHandler(ServerHandler.ACTION_SET_RESCHEDULING, data);
    addCallQuery.execute();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
};
Thread thread = new Thread(null,
backgroundProcess, "Background");
thread.start();
CalendarHandler.addEvent(context, numberToSend,
callStartTimeMillis, timeToSetMillis[0]);
SMSHandler.sendSMS(context, numberToSend,
textToSend.getText().toString());
PhoneHandler.endRingingCall(context,
numberToSend);

try{
    ((LaterCallsActivity)
parentObject).onResume();
} catch (Exception e) {
    e.printStackTrace();
}
removePCL();
}
}
}
});

Button buttonWait15 =
vglayout.findViewById(R.id.bubble_button_wait15);
buttonWait15.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Log.i("LOOK HERE: PCL", "+15 минут");
        timeToSetMillis[0] = callStartTimeMillis + 900000;
        String[] hms =
CalendarHandler.getTimeStringFromLong(timeToSetMillis[0],
"HH:mm").split(":");
        timePicker.setHour(Integer.valueOf(hms[0]));
        timePicker.setMinute(Integer.valueOf(hms[1]));

        textToSend.setText(context.getResources().getString(R.string.textSMS,
CalendarHandler.getTimeStringFromLong(timeToSetMillis[0], "dd.MM.y HH:mm")));
        if (buttonSend.getText().toString().equals("Далее")) {
            isConfirmed[0] = false;
            buttonSend.callOnClick();
        }
        isConfirmed[0] = false;
        buttonSend.callOnClick();
    }
});

```

```

        Button buttonWait30 =
        vgLayout.findViewById(R.id.bubble_button_wait30);
        buttonWait30.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.i("LOOK HERE: PCL", "+30 минут");
                timeToSetMillis[0] = callStartTimeMillis + 1800000;
                String[] hms =
                CalendarHandler.getTimeStringFromLong(timeToSetMillis[0],
                "HH:mm").split(":");
                timePicker.setHour(Integer.valueOf(hms[0]));
                timePicker.setMinute(Integer.valueOf(hms[1]));

                textToSend.setText(context.getResources().getString(R.string.textSMS,
                CalendarHandler.getTimeStringFromLong(timeToSetMillis[0], "dd.MM.y HH:mm")));
                if (buttonSend.getText().toString().equals("Далее")) {
                    isConfirmed[0] = false;
                    buttonSend.callOnClick();
                }
                isConfirmed[0] = false;
                buttonSend.callOnClick();
            }
        });

        Button buttonWait1 = vgLayout.findViewById(R.id.bubble_button_wait1);
        buttonWait1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.i("LOOK HERE: PCL", "+1 час");
                timeToSetMillis[0] = callStartTimeMillis + 3600000;
                String[] hms =
                CalendarHandler.getTimeStringFromLong(timeToSetMillis[0],
                "HH:mm").split(":");
                timePicker.setHour(Integer.valueOf(hms[0]));
                timePicker.setMinute(Integer.valueOf(hms[1]));

                textToSend.setText(context.getResources().getString(R.string.textSMS,
                CalendarHandler.getTimeStringFromLong(timeToSetMillis[0], "dd.MM.y HH:mm")));
                if (buttonSend.getText().toString().equals("Далее")) {
                    isConfirmed[0] = false;
                    buttonSend.callOnClick();
                }
                isConfirmed[0] = false;
                buttonSend.callOnClick();
            }
        });

        buttonReturn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String brt = buttonReturn.getText().toString();
                if (brt.equals("Назад")) {
                    lLayout.removeView(timePicker);
                    lLayout.addView(datePicker, 1);
                    buttonSend.setText("Далее");
                    buttonReturn.setText("Отменить");
                    Log.i("LOOK HERE: PCL", "Назад");
                } else if (brt.equals("Отменить")) {
                    removePCL();
                }
            }
        });
    }
}

```

```

        public static int[] getActivitySize(Context context) {
            WindowManager windowManager = (WindowManager)
context.getSystemService(Context.WINDOW_SERVICE);
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
                WindowMetrics windowMetrics =
windowManager.getCurrentWindowMetrics();
                Insets insets =
windowMetrics.getWindowInsets().getInsetsIgnoringVisibility(WindowInsets.Type
.systemBars());
                return new int[] {windowMetrics.getBounds().width() - insets.left
- insets.right, windowMetrics.getBounds().height() - insets.top -
insets.bottom};
            } else {
                DisplayMetrics displayMetrics = new DisplayMetrics();
                windowManager.getDefaultDisplay().getMetrics(displayMetrics);
                return new int[] {displayMetrics.widthPixels,
displayMetrics.heightPixels};
            }
        }

        //Метод удаления текстового пузыря
        public void removePCL() {
            try {
                windowMan.removeView(vgLayout); //Удаляем раздутый макет из окна
                isShown = false;
                Log.i("LOOK HERE: PCL", "PCL has been removed");
            } catch (IllegalArgumentException e) {
                Log.i("LOOK HERE: PCL", "PCL has not found");
            } catch (NullPointerException e) {
                Log.i("LOOK HERE: PCL", "windowMan has not found");
            }
        }
    }
}

```

ServerHandler

```

package com.shilina.project_x;

import android.os.AsyncTask;
import android.util.Log;
import android.widget.Toast;

import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import static java.lang.Boolean.valueOf;

class ServerHandler extends AsyncTask<Void, Void, String>
{
    public static final String HOST = "192.168.0.101";
    public static final String DB_LOGIN = "asd";
    public static final String DB_PASS = "asd";
}

```

```

public static final String ACTION_LOGIN = "login";
public static final String ACTION_SIGNUP = "signup";
public static final String ACTION_SET_PREMIUM = "set_premium";
public static final String ACTION_SET_SETTING = "set_setting";
public static final String ACTION_CHANGE_PASS = "change_pass";
public static final String ACTION_GET_RESCHEDULING = "get_rescheduling";
public static final String ACTION_SET_RESCHEDULING = "set_rescheduling";
public static final String ACTION_DELETE_CALL = "delete_call";

HashMap<String, String> data;
String action;
String resultString = null;

public ServerHandler(String action, HashMap<String, String> data) {
    this.action = action;
    this.data = data;
}

@Override
protected void onPreExecute()
{
    super.onPreExecute();
}

@Override
protected String doInBackground(Void... params) {
    try {
        String myURL = "http://" + HOST + "/?action=" + action;
        String myData = "host=" + HOST + "&db_login=" + DB_LOGIN +
"&db_pass=" + DB_PASS;

        for (String key : data.keySet()) {
            String value = data.get(key);
            myData += "&" + key + "=" + value;
        }

        byte[] streamData = null;
        InputStream inputStream = null;

        Log.i("LOOK HERE: SH", "Connecting to server...");
        URL url = new URL(myURL);
        HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
        conn.setReadTimeout(10000);
        conn.setConnectTimeout(15000);
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Connection", "Keep-Alive");
        conn.setRequestProperty("Content-Type", "application/x-www-form-
urlencoded");
        conn.setRequestProperty("Content-Length", "" +
myData.getBytes().length);
        conn.setDoOutput(true);
        conn.setDoInput(true);

        //Передаем данные на сервер
        streamData = myData.getBytes("UTF-8");
        OutputStream os = conn.getOutputStream();
        os.write(streamData);
        os.flush();
        os.close();
        conn.connect();
        int responseCode = conn.getResponseCode();
    }
}

```

```

        //Получаем ответ с сервера
        Log.i("LOOK HERE: SH", "Reading server response...");
        streamData = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        if (responseCode == HttpURLConnection.HTTP_OK) {
            inputStream = conn.getInputStream();
            byte[] buffer = new byte[8192];
            // Далее так читаем ответ
            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                baos.write(buffer, 0, bytesRead);
            }
            streamData = baos.toByteArray();
            resultString = new String(streamData, "UTF-8"); // сохраняем
        в переменную ответ сервера, у нас "OK"
        } else {
            resultString = "Ошибка сервера";
        }
        conn.disconnect();
    } catch (Exception e) {
        resultString = "Exception: " + e.getMessage();
    }
    Log.i("LOOK HERE: SH", "OTBET: " + resultString);
    return resultString;
}

@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
}

public static String isErrored (String str) {
    try {
        JSONObject result_obj = new JSONObject(str);
        if ("1".equals(result_obj.getString("error"))) {
            String error = result_obj.getString("error_text");
            Log.i("LOOK HERE: SH", "Query error: " + error);
            return error;
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    Log.i("LOOK HERE: SH", "Query succeed");
    return null;
}
}

```

SettingsActivity

```

package com.shilina.project_x;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;

```

```

import android.widget.TextView;
import android.widget.Toast;

import java.util.Arrays;
import java.util.Calendar;
import java.util.HashMap;

public class SettingsActivity extends DrawerActivity {

    public static final String className =
Thread.currentThread().getStackTrace()[2].getClassName();
    public static final String SP_FILE = "Accaunt";
    public static final String SP_THEME = "Theme";
    public static final String[] SP_THEMES = new String[] {"Shark",
"Raccoon", "Panther"};
    public static final String SP_MODE_IN = "ModeIn";
    public static final String[] SP_MODES_IN = new String[] {"Hand", "Auto"};
    public static final String SP_MODE_OUT = "ModeOut";
    public static final String[] SP_MODES_OUT = new String[] {"Hand", "Off"};
    public static final String SP_USER = "User";
    public static final String SP_PREMIUM = "Premium"; //bool
    SharedPreferences sp_settings;
    SharedPreferences.Editor prefEditor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SettingsActivity.chooseTheme(this);
        setContentView(R.layout.activity_settings);
        super.drawerLayout = findViewById(R.id.drawer_layout);
        super.className = className;
        setNickname();
        Log.i("LOOK HERE: SettingsActivity", "Opened activity is: " +
className);

        TextView textById = findViewById(R.id.nameOfWindow);
        textById.setText("Настройки");
        sp_settings = getSharedPreferences(SP_FILE, MODE_PRIVATE);
        prefEditor = sp_settings.edit();

        //Обработка смены темы
        String sp_theme = sp_settings.getString(SP_THEME, SP_THEMES[0]);
        String curTheme = sp_theme;
        Spinner spinnerThemes = findViewById(R.id.spinnerTheme);
        ArrayAdapter<?> adapter = ArrayAdapter.createFromResource(this,
R.array.themes, android.R.layout.simple_spinner_item);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_item);
        spinnerThemes.setAdapter(adapter);

        spinnerThemes.setSelection(Arrays.asList(SP_THEMES).indexOf(sp_theme));
        spinnerThemes.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View
itemSelected, int selectedItemPosition, long selectedId) {
                String newTheme = SP_THEMES[selectedItemPosition];
                Log.i("LOOK HERE: SettingsActivity", "Cur Theme is: " +
curTheme + "\nNew Theme is: " + newTheme);
                if (!curTheme.equals(newTheme)) {
                    if (isPremium(getApplicationContext())) {
                        prefEditor.putString(SP_THEME, newTheme);
                        prefEditor.apply();

                        Runnable backgroundProcess = new Runnable() {

```



```

        public void run() {
            try {
                HashMap<String, String> data = new
HashMap<String, String>() {{
                    put("email",
SettingsActivity.getUser(getApplicationContext()));
                    put("device_name",
Settings.Secure.getString(getApplicationContext().getContentResolver(),
Settings.Secure.ANDROID_ID));
                    put("setting_name", "color_logo");
                    put("value", sp_theme);
                }};
                ServerHandler setQuery = new
ServerHandler(ServerHandler.ACTION_SET_SETTING, data);
                setQuery.execute();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        Thread thread = new Thread(null, backgroundProcess,
"Background");
        thread.start();
        startActivity(getIntent());
    } else {
        spinnerThemes.setSelection(0);
        Toast.makeText(getApplicationContext(), "Чтобы
сменить тему, купите премиум", Toast.LENGTH_SHORT).show();
    }
}

    public void onNothingSelected(AdapterView<?> parent) {
    }
});

//Обработка смены режима вовремя мероприятий
RadioGroup rgModeIn = findViewById(R.id.modeIn);
Integer[] intModesIn = new Integer[2];
for (int i=0; i<rgModeIn.getChildCount(); i++) {
    RadioButton rbModeIn = (RadioButton) rgModeIn.getChildAt(i);
    intModesIn[i] = rbModeIn.getId();
}
String sp_mode_in = sp_settings.getString(SP_MODE_IN,
SP_MODES_IN[0]);
String curModeIn= sp_mode_in;

rgModeIn.check(intModesIn[Arrays.asList(SP_MODES_IN).indexOf(curModeIn)]);
rgModeIn.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
    public void onCheckedChanged(RadioGroup rg, int newModeInID) {
        String newModeIn =
SP_MODES_IN[Arrays.asList(intModesIn).indexOf(newModeInID)];
        Log.i("LOOK HERE: SettingsActivity", "Cur ModeIn is: " +
curModeIn + "\nNew ModeIn is: " + newModeIn);
        prefEditor.putString(SP_MODE_IN, newModeIn);
        prefEditor.apply();

        Runnable backgroundProcess = new Runnable() {
            public void run() {
                try {
                    HashMap<String, String> data = new
HashMap<String, String>() {{
                        put("email",
SettingsActivity.getUser(getApplicationContext()));

```

```

        put("device_name",
Settings.Secure.getString(getApplicationContext().getContentResolver(),
Settings.Secure.ANDROID_ID));
        put("setting_name", "rescheduling_in_event");
        put("value", newModeIn);
    });
    ServerHandler setQuery = new
ServerHandler(ServerHandler.ACTION_SET_SETTING, data);
    setQuery.execute();
    } catch (Exception e) {
        e.printStackTrace();
    }
    };
    Thread thread = new Thread(null, backgroundProcess,
"Background");
    thread.start();
    }
    });

    //Обработка смены режима вне мероприятий
    RadioGroup rgModeOut = findViewById(R.id.modeOut);
    Integer[] intModesOut = new Integer[2];
    for (int i=0; i<rgModeOut.getChildCount(); i++) {
        RadioButton rbModeOut = (RadioButton) rgModeOut.getChildAt(i);
        intModesOut[i] = rbModeOut.getId();
    }
    String sp_mode_out = sp_settings.getString(SP_MODE_OUT,
SP_MODES_OUT[0]);
    String curModeOut= sp_mode_out;

    rgModeOut.check(intModesOut[Arrays.asList(SP_MODES_OUT).indexOf(curModeOut)])
    ;

    rgModeOut.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
        public void onCheckedChanged(RadioGroup rg, int newModeOutID) {
            String newModeOut =
SP_MODES_OUT[Arrays.asList(intModesOut).indexOf(newModeOutID)];
            Log.i("LOOK HERE: SettingsActivity", "Cur ModeOut is: " +
curModeOut + "\nNew ModeOut is: " + newModeOut);
            prefEditor.putString(SP_MODE_OUT, newModeOut);
            prefEditor.apply();

            Runnable backgroundProcess = new Runnable() {
                public void run() {
                    try {
                        HashMap<String, String> data = new
HashMap<String, String>() {{
                            put("email",
SettingsActivity.getUser(getApplicationContext()));
                            put("device_name",
Settings.Secure.getString(getApplicationContext().getContentResolver(),
Settings.Secure.ANDROID_ID));
                            put("setting_name",
"rescheduling_out_event");
                            put("value", newModeOut);
                        }};
                        ServerHandler setQuery = new
ServerHandler(ServerHandler.ACTION_SET_SETTING, data);
                        setQuery.execute();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }
        }
    }
}

```

```

        };
        Thread thread = new Thread(null, backgroundProcess,
"Background");
        thread.start();
    }
});

}

    public void onPasswordChangeClick(View view) {
        String pass1 = ((EditText)
findViewById(R.id.settings_password)).getText().toString();
        String pass2 = ((EditText)
findViewById(R.id.settings_password_check)).getText().toString();
        if (pass1.equals(pass2)) {
            Runnable backgroundProcess = new Runnable() {
                public void run() {
                    try {
                        HashMap<String, String> data = new HashMap<String,
String>() {{
                            put("email",
SettingsActivity.getUser(getApplicationContext()));
                            put("new_pass", pass1);
                        }};
                        ServerHandler passQuery = new
ServerHandler(ServerHandler.ACTION_CHANGE_PASS, data);
                        passQuery.execute();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            };
            Thread thread = new Thread(null, backgroundProcess,
"Background");
            thread.start();
            Toast.makeText(getApplicationContext(), "Пароль был изменен",
Toast.LENGTH_SHORT).show();
            Log.i("LOOK HERE: SettingsActivity", "Password was changed");
        } else {
            Toast.makeText(getApplicationContext(), "Пароли не совпадают",
Toast.LENGTH_SHORT).show();
            Log.i("LOOK HERE: SettingsActivity", "Password was NOT changed");
        }
    }

    public void onResetClick(View view){
        setAllDefault(getApplicationContext());
        startActivity(getIntent());
    }

    public void onPermissionsClick(View view) {
        Intent intent = new Intent(getApplicationContext(),
PermissionsActivity.class);
        intent.putExtra("calling-activity", className);
        startActivity(intent);
    }

    public static void chooseTheme(Context context){
        SharedPreferences sp_settings = context.getSharedPreferences(SP_FILE,
MODE_PRIVATE);
        String sp_theme = sp_settings.getString(SP_THEME, SP_THEMES[0]);
        switch (sp_theme) {
            case "Shark":

```

```

        context.setTheme(R.style.Theme_Project_X);
        break;
    case "Raccoon":
        context.setTheme(R.style.Theme_Project_X_enot);
        break;
    case "Panther":
        context.setTheme(R.style.Theme_Project_X_pantera);
        break;
    }
    Log.i("LOOK HERE: SettingsActivity", "Cur Theme is: " + sp_theme);
}

public static void setUser(Context context, String user) {
    SharedPreferences sp_settings = context.getSharedPreferences(SP_FILE,
MODE_PRIVATE);
    SharedPreferences.Editor prefEditor = sp_settings.edit();
    prefEditor.putString(SP_USER, user);
    prefEditor.apply();
    Log.i("LOOK HERE: SettingsActivity", "Authorise as: " + user);
}

public static String getUser(Context context) {
    SharedPreferences sp_settings = context.getSharedPreferences(SP_FILE,
MODE_PRIVATE);
    String sp_user = sp_settings.getString(SP_USER, null);
    Log.i("LOOK HERE: SettingsActivity", "Authorised as: " + sp_user);
    return sp_user;
}

public static boolean isAuthorised(Context context){
    if (getUser(context) == null){
        Log.i("LOOK HERE: SettingsActivity", "NOT Authorised");
        return false;
    } else {
        Log.i("LOOK HERE: SettingsActivity", "Authorised");
        return true;
    }
}

public static void setPremium(Context context, boolean value) {
    SharedPreferences sp_settings = context.getSharedPreferences(SP_FILE,
MODE_PRIVATE);
    SharedPreferences.Editor prefEditor = sp_settings.edit();
    prefEditor.putBoolean(SP_PREMIUM, value);
    prefEditor.apply();
    Log.i("LOOK HERE: SettingsActivity", "Premium now: " + value);
}

public static boolean isPremium(Context context) {
    SharedPreferences sp_settings = context.getSharedPreferences(SP_FILE,
MODE_PRIVATE);
    boolean sp_premium = sp_settings.getBoolean(SP_PREMIUM, false);
    Log.i("LOOK HERE: SettingsActivity", "Premium is: " + sp_premium);
    return sp_premium;
}

public static void setAllDefault(Context context) {
    SharedPreferences sp_settings = context.getSharedPreferences(SP_FILE,
MODE_PRIVATE);
    SharedPreferences.Editor prefEditor = sp_settings.edit();
    prefEditor.putString(SP_THEME, SP_THEMES[0]);
    prefEditor.putString(SP_MODE_IN, SP_MODES_IN[0]);
    prefEditor.putString(SP_MODE_OUT, SP_MODES_OUT[0]);
    prefEditor.apply();
}

```

```

        String androidId =
Settings.Secure.getString(context.getContentResolver(),
Settings.Secure.ANDROID_ID);
        Log.i("LOOK HERE: SettingsActivity", "Settings are set to default");
    }

}

```

SignUpActivity

```

package com.shilina.project_x;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.provider.Settings;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.regex.Pattern;

import static java.lang.String.valueOf;

public class SignUpActivity extends AppCompatActivity {

    EditText email;
    EditText pass1;
    EditText pass2;
    EditText phone;
    TextView textAlert;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SettingsActivity.chooseTheme(this);
        setContentView(R.layout.activity_sign_up);
        email = (EditText) findViewById(R.id.email);
        pass1 = (EditText) findViewById(R.id.enterPassword);
        pass2 = (EditText) findViewById(R.id.checkPassword);
        phone = (EditText) findViewById(R.id.phone_number);
        textAlert = (TextView) findViewById(R.id.textAlertSign);
    }

    private void EnterErrors() {
        //Is correct Edit texts?
    }

    public void onSingUpButtonClick(View view) {
        pass1.setBackgroundColor(Color.red(1));
        pass2.setBackgroundColor(Color.red(1));
    }
}

```

```

String email_in = email.getText().toString();
String pass1_in = pass1.getText().toString();
String pass2_in = pass2.getText().toString();
String phone_in = phone.getText().toString();
String android_id = Settings.Secure.getString(getContentResolver(),
Settings.Secure.ANDROID_ID);

List<String> errors = new ArrayList<String>();
if (!email_in.matches("\\w+@gmail\\.com")) {
    errors.add("Почта не принадлежит домену gmail.com");
}
if (email_in.length() >= 50) {
    errors.add("Длина почты слишком большая");
}
if (!Pattern.matches("^((8|\\+7)[\\- ])?(\\((?\\d{3}\\))?(\\-
]?)?(\\d\\- ]{7,10})$", (phone_in))) {
    errors.add("Неверный формат номера телефона");
}
if (pass1_in.length() > 50 || pass1_in.length() < 1) {
    errors.add("Пароль должен содержать от 1 до 50 символов");
}
if (!pass1_in.equals(pass2_in)) {
    errors.add("Пароли не совпадают");
}

if (errors.size() > 0) {
    textAlert.setVisibility(View.VISIBLE);
    textAlert.setText(errors.get(0));
    return;
} else {
    Runnable backgroundProcess = new Runnable() {
        public void run() {
            try {
                Context context = getApplicationContext();
                HashMap<String, String> data = new HashMap<String,
String>() {{
                    put("phone", phone_in);
                    put("email", email_in);
                    put("pass", pass1_in);
                    put("android_id", android_id);
                }};
                ServerHandler signupQuery = new
ServerHandler(ServerHandler.ACTION_SIGNUP, data);
                signupQuery.execute();
                String responseString = signupQuery.get();
                JSONObject responseJSON = new
JSONObject(responseString);
                if (ServerHandler.isErrored(responseString) == null)
                {
                    new Handler(Looper.getMainLooper()).post(new
Runnable() {
                        public void run() {
                            Toast.makeText(context, "Регистрация
успешна", Toast.LENGTH_SHORT).show();
                        }
                    });
                    SettingsActivity.setUser(context,
data.get("email"));
                    SettingsActivity.setPremium(context, false);
                    Intent intent = new Intent(context,
LaterCallsActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                    context.startActivity(intent);
                }
            }
        }
    };
    Thread thread = new Thread(backgroundProcess);
    thread.start();
}
}

```

```

        } else {
            new Handler(Looper.getMainLooper()).post(new
Runnable() {
                public void run() {
                    try {
                        Toast.makeText(context, "Ошибка
регистрации: " + responseJSON.getString("error_text"),
Toast.LENGTH_SHORT).show();
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            });
        }
    } catch (Exception e) {
    }
}
};
Thread thread = new Thread(null, backgroundProcess, "Background");
thread.start();
}
}
}

```

SMSHandler

```

package com.shilina.project_x;

import android.content.Context;
import android.telephony.SmsManager;
import android.util.Log;
import android.widget.Toast;

public class SMSHandler {

    //Метод отправки СМС
    public static void sendSMS(Context context, String recipient, String
message) {
        try {
            SmsManager smsMan = SmsManager.getDefault();
            smsMan.sendTextMessage(recipient, null, message, null, null);
            Log.i("LOOK HERE: SMSHandler", "SMS was sent");
        } catch (Exception e) {
            Log.i("LOOK HERE: SMSHandler", "SMS wasn't sent");
        }
    }
}

```

StatusActivity

```

package com.shilina.project_x;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;

```

```

import android.text.Html;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.Calendar;
import java.util.HashMap;

public class StatusActivity extends DrawerActivity{

    private AlertDialog.Builder dialogBuilder;
    private AlertDialog dialog;
    private EditText pay_number, pay_month, pay_year, pay_name, pay_cvc;
    private Button pay_cancel, pay_buy;
    private TextView textAlert;

    public void createPayMenuDialog() {
        dialogBuilder = new AlertDialog.Builder(this);
        final View pay_view =
getLayoutInflater().inflate(R.layout.activity_pay, null);
        pay_number = (EditText) pay_view.findViewById(R.id.pay_number);
        pay_month = (EditText) pay_view.findViewById(R.id.pay_month);
        pay_year = (EditText) pay_view.findViewById(R.id.pay_year);
        pay_name = (EditText) pay_view.findViewById(R.id.pay_name);
        pay_cvc = (EditText) pay_view.findViewById(R.id.pay_cvc);

        pay_buy = (Button) pay_view.findViewById(R.id.buy);
        pay_cancel = (Button) pay_view.findViewById(R.id.pay_cancel);

        textAlert = (TextView) pay_view.findViewById(R.id.textAlert);

        dialogBuilder.setView(pay_view);
        dialog = dialogBuilder.create();
        dialog.show();

        pay_cancel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //define cancel button here!
                dialog.dismiss();
            }
        });

        pay_buy.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if (pay_number.getText().toString().length() < 16 ||
pay_month.getText().toString().length() < 2 ||
                pay_cvc.getText().toString().length() < 3 ||
pay_year.getText().toString().length() < 2 ||
                pay_name.getText().toString() == "") {
                    textAlert.setVisibility(View.VISIBLE);
                    return;
                }
            }
        });
    }
}

```



```

        int month = Integer.parseInt(pay_month.getText().toString());
        int year = Integer.parseInt(pay_year.getText().toString());

        if (month < 1 || month > 12 || year < 21)
        {
            textAlert.setVisibility(View.VISIBLE);
        } else {

            Log.i("LOOK HERE: SA", "Purchased premium ");
            SettingsActivity.setPremium(getApplicationContext(),
true);

            Runnable backgroundProcess = new Runnable() {
                public void run() {
                    try {
                        Context context = getApplicationContext();
                        HashMap<String, String> data = new
HashMap<String, String>() {{
                            put("email",
SettingsActivity.getUser(context));
                            put("summ", "250");
                            put("pay_date",
CalendarHandler.getTimeStringFromDate(Calendar.getInstance().getTime(),
"dd.MM.yy"));
                            put("pay_time",
CalendarHandler.getTimeStringFromDate(Calendar.getInstance().getTime(),
"HH:mm"));
                        }};
                        ServerHandler premiumQuery = new
ServerHandler(ServerHandler.ACTION_SET_PREMIUM, data);
                        premiumQuery.execute();
                        String responseString = premiumQuery.get();
                        JSONObject responseJSON = new
JSONObject(responseString);
                        if (ServerHandler.isErrored(responseString)
== null) {
                            new
Handler(Looper.getMainLooper()).post(new Runnable() {
                                public void run() {
                                    Toast.makeText(context, "Премии
куплен", Toast.LENGTH_SHORT).show();
                                }
                            });
                            startActivity(getIntent());
                        } else {
                            new
Handler(Looper.getMainLooper()).post(new Runnable() {
                                public void run() {
                                    try {
                                        Toast.makeText(context,
"Ошибка покупки: " + responseJSON.getString("error_text"),
Toast.LENGTH_SHORT).show();
                                    } catch (JSONException e) {
                                        e.printStackTrace();
                                    }
                                }
                            });
                        }
                    } catch (Exception e) {
                    }
                }
            };
        }
    };

```

```

        Thread thread = new Thread(null, backgroundProcess,
"Background");
        thread.start();
    }
}

});

}

    public static final String className =
Thread.currentThread().getStackTrace()[2].getClassName();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    SettingsActivity.chooseTheme(this);
    if (!SettingsActivity.isPremium(getApplicationContext())) {
        setContentView(R.layout.activity_status);
    } else {
        setContentView(R.layout.activity_status_prem);
    }

    super.drawerLayout = findViewById(R.id.drawer_layout);
    super.className = className;
    setNickname();
    Log.i("LOOK HERE: SA", "Opened activity is: " + className);

    TextView textById = findViewById(R.id.nameOfWindow);
    textById.setText("Мой статус");
}

public void onPurchaseClick(View view) {
    createPayMenuDialog();
}
}

```

ПРИЛОЖЕНИЕ В. ЛИСТИНГ КОДА РАЗРАБОТКИ БД И СЕРВЕРА

Файл .sql в папке DB

caller_create_db

```
CREATE DATABASE Caller;
USE Caller;

CREATE TABLE Users
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    e_mail VARCHAR(50) NOT NULL UNIQUE,
    u_password VARCHAR(50) NOT NULL,
    premium BOOL NOT NULL
);

CREATE TABLE Numbers
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    phone_number VARCHAR(20) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES Users (id)
);

CREATE TABLE Devices
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    device_name NVARCHAR (50) NOT NULL,
    color_logo NVARCHAR (20) NOT NULL DEFAULT 'Shark',
    rescheduling_in_event NVARCHAR (50) NOT NULL DEFAULT 'Auto',
    rescheduling_out_event NVARCHAR (50) NOT NULL DEFAULT 'Auto'
);

CREATE TABLE UserDevice
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    device_id INT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES Users (id),
    FOREIGN KEY (device_id) REFERENCES Devices (id)
);

CREATE TABLE Calls
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    recipient_number VARCHAR(20) NOT NULL,
    call_date_time VARCHAR(20) NOT NULL,
    callback_date_time VARCHAR(20) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES Users (id)
);

CREATE TABLE Payments
(
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```

user_id      INT NOT NULL,
summ DECIMAL(5,2) NOT NULL,
pay_date_time DATETIME NOT NULL,
FOREIGN KEY (user_id) REFERENCES Users (id)
);

```

Файлы .php в папке SRC\Project_X_Server

index

```

<?php
require_once 'initialization.php';
$data = $_POST;
$service_root_path = "";
initialize($data['host'], $data['db_login'], $data['db_pass'],
$service_root_path);
$action = $_GET['action'];
//echo "action: ".$action."\n";

switch ($action)
{
    case "login":
        $scr = '/process/login.php';
        break;
    case "signup":
        $scr = '/process/sign_up.php';
        break;
    case "set_premium":
        $scr = '/process/set_premium.php';
        break;
    case "set_rescheduling":
        $scr = '/process/set_rescheduling.php';
        break;
    case "get_rescheduling":
        $scr = '/process/get_rescheduling.php';
        break;
    case "set_setting":
        $scr = '/process/set_setting.php';
        break;
    case "change_pass":
        $scr = '/process/change_pass.php';
        break;
    case "delete_call":
        $scr = '/process/delete_call.php';
        break;
    case "test":
        $scr = '/test.php';
        break;
    default:
        echo "Неверный адрес страницы";
        break;
}

require_once $service_root_path.$scr;

```

initialization

```

<?php
// Здесь процессы, которые необходимо выполнять при каждом обращении к
серверу
function initialize($host, $db_login, $db_pass, &$service_root_path)

```

```

{
    require "libs/rb-mysql.php";
    R::setup( 'mysql:host='.$host.';dbname=caller', $db_login, $db_pass );

    session_start();

    // Задаём корневую директорию для подключения файлов
    $service_root_path = dirname(__FILE__)."/..";
    $service_root_path = realpath($service_root_path);
    $service_root_path = str_replace("\\", "/", $service_root_path);
    $service_root_path = $service_root_path."/Project_X_Server";

    // Задаём, где искать файлы при подключении
    $include_path_separator = PATH_SEPARATOR;
    $include_path = $service_root_path;
    $include_path_dirs = array(
        "process",
    );
    foreach ($include_path_dirs as $dirname) {
        $include_path .=
    "{$include_path_separator}{$service_root_path}/{$dirname}";
    }
    set_include_path($include_path);
}

function vardump($var) {
    echo '<pre>';
    var_dump($var);
    echo '</pre>';
}

```

test

```

<?php
echo "return text 123 ".$_POST["name"].var_dump($_GET);

```

errors.html

```

<br />
<b>Warning</b>: require_once(/process/login.php): failed to open stream:
No such file or directory in
<b>D:\OpenServerPanel\OpenServer\domains\first_common_project\SRC\Project_X_S
erver\index.php</b> on line <b>39</b><br />
<br />
<b>Fatal error</b>: require_once(): Failed opening required
'/process/login.php'
(include_path='D:/OpenServerPanel/OpenServer/domains/first_common_project/SRC
/Project_X_Server;D:/OpenServerPanel/OpenServer/domains/first_common_project/
SRC/Project_X_Server/process') in
<b>D:\OpenServerPanel\OpenServer\domains\first_common_project\SRC\Project_X_S
erver\index.php</b> on line <b>39</b><br />

```

Файлы .php в папке SRC\Project_X_Server\process

change_pass

```

<?php
/*
* Обработка смены пароля
* Приходят данные:
* email - почта пользователя

```

```

* new_pass - новый пароль пользователя
*
* Возвращает:
* JSON с полями:
* * error: 1 или 0
* * error_text: пустая строка или текст ошибки
*/

$data = $_POST;
$errors = array();
$json_array = array(
    "error" => 1,
    "error_text" => "Неотлаживаемая ошибка",
);

$user = R::findOne('users', 'e_mail = ?', array($data['email']));
if ($user)
{
    $user -> u_password = $data["new_pass"];

    R::store($user);

    $json_array["error"] = 0;
    $json_array["error_text"] = "";
}
else $errors[] = 'Пользователь с такой почтой не найден \(._\)';

if(!empty($errors))
{
    $json_array["error"] = 1;
    $json_array["error_text"] = $errors[0];
}

echo json_encode($json_array);

```

delete_call

```

<?php
/*
* Обработка удаления звонка
* Приходят данные:
* id_call - ид звонка, который нужно удалить
*
* Возвращает:
* JSON с полями:
* * error: 1 или 0
* * error_text: пустая строка или текст ошибки
*/

$data = $_POST;
$errors = array();
$json_array = array(
    "error" => 1,
    "error_text" => "Неотлаживаемая ошибка",
);

$call = R::findOne('calls', 'id = ?', array($data['id_call']));
if ($call)
{
    R::trash($call);

    $json_array["error"] = 0;
    $json_array["error_text"] = "";
}
else $errors[] = 'Звонок с таким id не найден \(._\)';

```

```

if(!empty($errors))
{
    $json_array["error"] = 1;
    $json_array["error_text"] = $errors[0];
}

```

```

echo json_encode($json_array);

```

get_rescheduling

```

<?php
/*
 * Получение данных о перенесённых звонках пользователя с сейчас и дальше
 * Приходят данные:
 * email - почта пользователя
 *
 * Возвращает:
 * JSON с полями:
 * * error: 1 или 0
 * * error_text: пустая строка или текст ошибки
 * * calls: {phone: date_time}, где calls - звонки, phone - номер на который
    нужно перезвонить, call_datetime - время звонка в UNIX-time,
    callback_datetime - время, когда нужно перезвонить в UNIX-time
 */
$data = $_POST;
$errors = array();
$json_array = array(
    "error" => 1,
    "error_text" => "Неотлаживаемая ошибка",
    "calls" => array()
);

$user = R::findOne('users', 'e_mail = ?', array($data['email']));
$user_id = $user->id;
if ($user)
{
    $calls = R::find('calls', 'user_id = ? AND callback_date_time >= ? ORDER
BY callback_date_time ASC', array($user_id, time()));
    if ($calls)
    {
        $i = 1;
        foreach ($calls as $call)
        {
            $json_array["calls"][$i]["id"] = $call->id;
            $json_array["calls"][$i]["phone"] = $call -> recipient_number;
            $json_array["calls"][$i]["call_datetime"] = $call ->
call_date_time;
            $json_array["calls"][$i]["callback_datetime"] = $call ->
callback_date_time;
            $i ++;
        }

        $json_array["error"] = 0;
        $json_array["error_text"] = "";
    }
    else $errors[] = 'Нет запланированных звонков';
}
else $errors[] = 'Пользователь с такой почтой не найден \ (._) r';

if(!empty($errors))
{
    $json_array["error"] = 1;
    $json_array["error_text"] = $errors[0];
}

```

```
}
```

```
echo json_encode($json_array);
```

login

```
<?php
```

```
/*
```

```
 * Обработка авторизации
```

```
 * Приходят данные:
```

```
 * email - почта пользователя
```

```
 * pass - пароль пользователя
```

```
 * android_id - идентификатор андроида
```

```
 *
```

```
 * Возвращает:
```

```
 * JSON с полями:
```

```
 * * error: 1 или 0
```

```
 * * error_text: пустая строка или текст ошибки
```

```
 * * user_premium: 1 или 0 - есть премиум у пользователя или нет
```

```
 */
```

```
$data = $_POST;
```

```
$errors = array();
```

```
$json_array = array(
```

```
    "error" => 1,
```

```
    "error_text" => "Неотлаживаемая ошибка",
```

```
    "user_premium" => 0
```

```
);
```

```
$user = R::findOne('users', 'e_mail = ?', array($data['email']));
```

```
if ($user)
```

```
{
```

```
    if ($data['pass'] == $user->u_password)
```

```
    {
```

```
        $json_array["error"] = 0;
```

```
        $json_array["error_text"] = "";
```

```
        $json_array["user_premium"] = $user->premium;
```

```
    }
```

```
    else $errors[] = 'Неверно введен пароль';
```

```
    $device = R::findOne('devices', 'device_name = ?',  
array($data['android_id']));
```

```
    if (!$device)
```

```
    {
```

```
        $device = R::dispense('devices');
```

```
        $device->device_name = $data['android_id'];
```

```
        $device->color_logo = "Shark";
```

```
        $device->rescheduling_in_event = 1;
```

```
        $device->rescheduling_out_event = 1;
```

```
        R::store($device);
```

```
    }
```

```
}
```

```
else $errors[] = 'Пользователь с такой почтой не найден \(\._.\)';
```

```
if(!empty($errors))
```

```
{
```

```
    $json_array["error"] = 1;
```

```
    $json_array["error_text"] = $errors[0];
```

```
}
```

```
echo json_encode($json_array);
```

set_premium


```

<?php
/*
 * Обработка получения премиума
 * Приходят данные:
 * email - почта пользователя
 * summ - сумма платежа
 * pay_date - дата платежа в формате дд.мм.гггг
 * pay_time - время платежа в формате чч:мм
 *
 * Возвращает:
 * JSON с полями:
 * * error: 1 или 0
 * * error_text: пустая строка или текст ошибки
 */
$data = $_POST;
$errors = array();
$json_array = array(
    "error" => 1,
    "error_text" => "Неотлаживаемая ошибка",
);

$user = R::findOne('users', 'e_mail = ?', array($data['email']));
if ($user)
{
    $date_time = $data["pay_date"]." ".$data["pay_time"];

    $user->premium = 1;
    R::store($user);

    $pay = R::dispense('payments');
    $pay -> user_id = $user->id;
    $pay -> summ = $data['summ'];
    $pay -> pay_date_time = strtotime($date_time);
    R::store($pay);

    $json_array["error"] = 0;
    $json_array["error_text"] = "";
}
else $errors[] = 'Пользователь с такой почтой не найден \(\._\)r';

if(!empty($errors))
{
    $json_array["error"] = 1;
    $json_array["error_text"] = $errors[0];
}

echo json_encode($json_array);

```

set_rescheduling

```

<?php
/*
 * Обработка переноса звонка
 * Приходят данные:
 * email - почта пользователя
 * recipient_number - номер входящего звонка
 * call_datetime - дата и время звонка в формате UNIX-time
 * callback_datetime - дата и время, на которую переносится звонок в формате
UNIX-time
 *
 * Возвращает:
 * JSON с полями:

```

```

* * error: 1 или 0
* * error_text: пустая строка или текст ошибки
*/
$data = $_POST;
$errors = array();
$json_array = array(
    "error" => 1,
    "error_text" => "Неотлаживаемая ошибка",
);

$user = R::findOne('users', 'e_mail = ?', array($data['email']));
if ($user)
{
    $call = R::dispense('calls');
    $call -> user_id = $user->id;
    $call -> recipient_number = $data['recipient_number'];
    $call -> call_date_time = (int)$data['call_datetime'];
    $call -> callback_date_time = (int)$data['callback_datetime'];
    R::store($call);

    $json_array["error"] = 0;
    $json_array["error_text"] = "";
}
else $errors[] = 'Пользователь с такой почтой не найден \(\._.\)';

if(!empty($errors))
{
    $json_array["error"] = 1;
    $json_array["error_text"] = $errors[0];
}

echo json_encode($json_array);

```

set_setting

```

<?php
/*
* Обработка изменения настройки устройства
* Приходят данные:
* email - почта пользователя
* device_name - идентификатор андроида
* setting_name - название настройки. Могут быть: color_logo,
rescheduling_in_event, rescheduling_out_event.
* value - значение настройки, которое необходимо установить
*
* Возвращает:
* JSON с полями:
* * error: 1 или 0
* * error_text: пустая строка или текст ошибки
*/
$data = $_POST;
//var_dump($data);
$errors = array();
$json_array = array(
    "error" => 1,
    "error_text" => "Неотлаживаемая ошибка",
);

$user = R::findOne('users', 'e_mail = ?', array($data['email']));
if ($user)
{
    $u_device = R::findOne('userdevice', 'user_id = ?', array($user -> id));
    if ($u_device)
    {

```

```

        $device = R::findOne('devices', 'device_name = ?',
array($data["device_name"]));
//      var_dump($device);
        if ($device)
        {
            switch ($data["setting_name"])
            {
                case "color_logo":
                    $device -> color_logo = $data["value"];
                    break;
                case "rescheduling_in_event":
                    $device -> rescheduling_in_event = $data["value"];
                    break;
                case "rescheduling_out_event":
                    $device -> rescheduling_out_event = $data["value"];
                    break;
            }

            R::store($device);

            $json_array["error"] = 0;
            $json_array["error_text"] = "";
        }
        else $errors[] = 'У данного пользователя нет устройства с этим
AndroidId \(._.)\r';
    }
    else $errors[] = 'У данного пользователя нет устройств \(._.)\r';
}
else $errors[] = 'Пользователь с такой почтой не найден \(._.)\r';

if(!empty($errors))
{
    $json_array["error"] = 1;
    $json_array["error_text"] = $errors[0];
}

echo json_encode($json_array);

```

sign_up

```

<?php
/*
 * Обработка регистрации
 * Создаются новые записи в таблицах users, numbers, user_device и devices. В
последнюю таблицу записываются стандартные настройки
 * Приходят данные:
 * email - почта пользователя
 * phone - номер телефона
 * pass - пароль пользователя
 * android_id - идентификатор андроида. Уникален и генерируется на стороне клиента.
 *
 * Возвращает:
 * JSON с полями:
 * * error: 1 или 0
 * * error_text: пустая строка или текст ошибки
 */

$data = $_POST;
$errors = array();
$json_array = array(
    "error" => 1,
    "error_text" => "Неотлаживаемая ошибка",
);

```

```

//var_dump($data);

if(R::count('users', "e_mail = ?", array($data['email'])) > 0) $errors[] =
'Пользователь с таким email уже существует';
if(R::count('numbers', "phone_number = ?", array($data['phone'])) > 0)
$errors[] = 'Пользователь с таким номером телефона уже существует';

if(empty($errors))
{
    //echo "Всё ок";
    $user = R::dispense('users');
    $user -> e_mail = $data['email'];
    $user -> u_password = $data['pass'];
    $user -> premium = 0;
    R::store($user);

    $number = R::dispense("numbers");
    $number -> phone_number = $data['phone'];
    $number -> user_id = $user -> id;
    R::store($number);

    $device = R::dispense("devices");
    $device -> device_name = $data['android_id'];
    $device -> color_logo = "Shark";
    $device -> rescheduling_in_event = 1;
    $device -> rescheduling_out_event = 1;
    R::store($device);

    $u_d = R::dispense("userdevice");
    $u_d -> user_id = $user -> id;
    $u_d -> device_id = $device -> id;
    R::store($u_d);

    $json_array["error"] = 0;
    $json_array["error_text"] = "";
}
else
{
    $json_array["error"] = 1;
    $json_array["error_text"] = $errors[0];
}
echo json_encode($json_array);

```