*Приложение*

*к заявке на предоставление*

*инжиниринговой услуги*

**Техническое задание**

**на услуги** **«Разработка управляющих программ для управления оборудованием»**

**1.Общие положения.**

*Информация скрыта* (далее – Заказчик) сформировало настоящее техническое задание на предоставление услуги «Разработка управляющих программ для управления оборудованием» в целях разработки мобильного приложения «Умный дом» предназначенного для управления техническими устройствами, их мониторинга и иного взаимодействия.

**Разработка мобильного приложения для выпускаемого прибора заказчика:**

Заказчик осуществляет деятельность по производству GSM сигнализаций и систем управления. В настоящее время разработан новый прибор «», для которого необходимо изготовить мобильное приложение для телефонов на платформах IOS и Android для управления новым прибором заказчика

**Объект предоставления услуги**: *Информация скрыта*. Разработка мобильного приложения для разработанного прибора Заказчика.

**2. Требования, предъявляемые к Исполнителю.**

Определения

**Online** – режим работы приложения при наличии подключения устройства к сети.

**API clien**t – компонент, отвечающий за вызов удаленных методов сервера (API).

**UI** – графический интерфейс взаимодействия пользователя с приложением.

**Action Sheet** – графический элемент, который отображает панель контекстных действий пользователя.

**Свайп** – ­​направленное скольжение по экрану. В основном используется для пролистывания списков, открытия экрана меню.

**Placeholder​­** – текст-­подсказка, отображаемый в полях ввода, пока это поле не имеет значения.

**IMEI** **(*International Mobile Equipment Identity*)** – ­​уникальный идентификатор прибора.

**Loadmask** –​­ графический элемент, блокирующий интерфейс приложения на определённое время и отображающий индикатор некоторого процесса.

**Валидация** – проверка указанных пользователем данных, при заполнении различных форм.

**%значение%**​­ – вывод на экране подставленного значения из определенного параметра или поля.

**Картинка по умолчанию​­** – картинка, отображаемая до получения с сервера актуальной картинки, либо отображаемая картинка, если картинка в серверных данных была не указана.

**Список** – элемент отображения данных в приложении. Обычно по ширине совпадает с шириной экрана. Пункты списка выводятся в виде горизонтальных элементов, по ширине совпадающих с шириной списка. Высота содержимого списка может быть больше высоты экрана. Для просмотра всех данных в таком случае нужно пролистать содержимое списка с помощью свайпа снизу-вверх по области экрана со списком.

**Пустой** **список** – если в каком-то списке нет данных для отображения, то будет выведена надпись: “Список пуст”, расположенная по центру области, отведенной для списка. Текст надписи может варьироваться в зависимости от контекста. Например, в случае вывода в списке результатов поиска сотрудников, выводится надпись: “Сотрудники не найдены” и т.п.

**Тап** ­​- нажатие на экран.

**Лонг Тап** - долгое нажатие на экран.

Поддерживаемые платформы и версии ОС

Поддержка iOS: Версия с 9 по 11

Поддержка Android: Версия с 4.4.x по 8.1

Поддерживаемые устройства: смартфоны, планшеты.

Поддерживаемая ориентация: Portrait.

Локализация приложения: русская (Ru-ru)

Адаптация под следующие размеры: 4”, 4.7”, 5.1”, 5.7”, 7.9”, 9.7”

Приемочное тестирование. Будет проходить на следующих устройствах:

Android:

* Asus ZenFone 2 ZE550ML (Android 5.0) 5.5 inch
* Moto X (Android 6.0) 4.95 inch
* Samsung Galaxy S6 SM-G920F (Android 6.0) 5.1 inch
* Nexus 5X (Android 7.x) 5.2 inch

iOS:

* Apple iPad Air 2 (iOS 8.4.1) 9.7 inch
* iPad mini 2 (iOS 9.3.1) 7.9 inch
* iPhone 5 (iOS 10.0.2) 4 inch
* iPhone 6 (iOS 10.1.1) 4.5 inch
* iPhone 6+ (iOS 10.1.1) 5.5 inch

Общая информация

Дизайн для мобильного приложения предоставляет Исполнитель.

Текстовое описание текущего документа является более приоритетным, чем элементы, представленные на рисунках.

Кроссплатформенная (с использованием языка программирования С#) разработка с использованием фреймворка Xamarin Android и Xamarin iOS (<https://www.xamarin.com>)​

В поля ввода пользователь может вводить любые символы, если не оговорено иное.

Описание SMS

Для взаимодействия с прибором используются следующие смс-команды:

Запрос состояния объекта

Исходящее смс-сообщение: ***S***

Входящее смс-сообщение: ***Охрана state***

***Вх-***{v}n

***{Name State}m***

***{Tnum=zttt}k***

***Пит nut***

***state*** - состояние режима “Охрана” ***(Вкл -*** включена, ***Выкл*** - выключена***)***

***{v}n*** - ***v***- состояние входа, повторяется ***n*** раз, где ***n*** - количество входов. Возможные значения для состояния входа (***v***) - 0 - выключен, 1 - включен, T - состояние тревоги.

***{Name State}m*** - ***Name*** - название выхода, ***State*** - состояние (возможные значения:ВКЛ - включен, ВЫКЛ - выключен), ***m*** - повторять столько раз, сколько выходов имеет прибор.

***{Tnum=zttt}k*** - ***num*** -номер датчика температуры***, z*** - знак температуры на термодатчике (+ или -), ***k*** - повторять столько раз, сколько термодатчиков подключено к прибору.

***Пит nut - nut*** - состояние питания ***(ОК*** – нормальное Питание от сети, ***ВЫКЛ*** – Основное питание отключено, работа от аккумулятора).

Поставить на охрану

Исходящее смс-сообщение: ***O1***

Входящее смс-сообщение: ***Охрана ВКЛ***

Снять с охраны

Исходящее смс-сообщение: ***O0***

Входящее смс-сообщение: ***Охрана ВЫКЛ***

Управление выходами

Исходящее смс-сообщение: ***VXY[Tttt]***

Х - номер выхода

Y - состояние выхода (0 - выключить, 1 - включить)

Необязательные параметры:

Т - код команды, означающий время включения.

ttt – время на которое нужно включить выход в секундах.

Входящее смс-сообщение: ***Name State***

Name - название выхода

State - состояние выхода - возможно три состояния:

ВКЛ - включен,

ВЫКЛ - выключен,

ИМП - импульсный режим (выход включен на время)

Управление датчиками температуры

Исходящее смс-сообщение: ***TNzttt***

N - номер термодатчика по которому будет выставляться температура

z - указатель знака температуры (+ плюсовая температура; - минусовая)

***ttt*** – значение температуры в °С.

Входящее смс-сообщение: ***TNуст=zttt***

N - номер термодатчика

z - указатель знака температуры (+ плюсовая температура; - минусовая)

***ttt*** – значение температуры в °С.

Запрос температуры

Исходящее смс-сообщение: ***T?***

Входящее смс-сообщение: ***TN=zttt***

N - номер термодатчика

z - указатель знака температуры (+ плюсовая температура; - минусовая)

***ttt*** – значение температуры в °С.

Включение/отключение проводного входа

Исходящее смс-сообщение: ***EXY***

***X*** - номер проводного входа

***Y*** - команда для включения/выключения:

0 - выключить

1- включить

Входящее смс-сообщение: **Вход N State**

N - номер входа

**State** - состояние входа:

**Вкл** - вход включен,

**Выкл** - вход выключен.

Включение/отключение беспроводного входа

Исходящее смс-сообщение: ***EWXY***

***X*** - номер беспроводного входа

***Y*** - команда для включения/выключения:

0 - выключить

1- включить

Входящее смс-сообщение: **Вход N State**

N - номер входа

**State** - состояние входа:

**Вкл** - вход включен,

**Выкл** - вход выключен.

Включение/отключение датчика температуры

Исходящее смс-сообщение: ***PXY***

***X*** - номер датчика температуры

***Y*** - команда для включения/выключения:

0 - выключить

1- включить

Входящее смс-сообщение:  **Т N State**

N - номер термодатчика

**State** - состояние датчика:

**Вкл** - включен,

**Выкл** - выключен.

Включение/отключение контроля 24 часа для входов

Исходящее смс-сообщение: Hxyzn

**x** - требуемое состояние для первого входа (0 - выключить контроль 24 часа, 1 - включить контроль 24 часа)

**y** - требуемое состояние для второго входа (0 - выключить контроль 24 часа, 1 - включить контроль 24 часа)

**z** - требуемое состояние для третьего входа (0 - выключить контроль 24 часа, 1 - включить контроль 24 часа)

**n** - требуемое состояние для четвертого входа (0 - выключить контроль 24 часа, 1 0 включить контроль 24 часа)

Входящее смс-сообщение: **Контр *24 Установлен***

Задание общих параметров

Исходящее смс-сообщение: Xabcdfghjklmnpq

**a** - уровень постановки. Возможные значения:

- 1 - высокий уровень

- 2 - низкий уровень

- 3 - TouchMemory / импульс

**b** - отправка смс при постановке на охрану (0 - не отправлять, 1 - отправлять)

**с** - отправка смс при снятии с охраны (0 - не отправлять, 1 - отправлять)

**d** - звуковой сигнал при постановке\снятии охраны (0-выключить, 1-включить)

**f** - контроль напряжения питания (0-выключить, 1-включить)

**g** - использовать внешний ИБП (0-выключить, 1-включить)

**h** - добавлять Время в SMS сообщении (0-выключить, 1-включить)

**j** - при входящем звонке включить микрофон (0-выключить, 1-включить)

**k** - при звонке выслать SMS состояния (0-выключить, 1-включить)

**l** - включать выход при постановке на охрану (0 – не включать выход, 1 - включить Реле1, 2 - включить Реле2, 3 - включить Выход1(ОК1), 4 – включить Выход2(ОК2)

**m** - при тревоге делать отправку SMS (0-выключить, 1-включить)

**n** - при тревоге делать дозвон (0-выключить, 1-включить)

**p** - тип индикатора состояния (1- светодиод, 2 – внешний индикатор)

**q** - отправлять SMS сообщения сразу на все номера (0-выключить, 1-включить)

1. Принципы работы приложения “ИПРО-7”

Мобильное приложение «ИПРО-7» для мобильных платформ представляет собой панель контроля и управления прибором сигнализации. Связь с приборами осуществляется посредством смс-сообщений, с помощью которых отправляются запросы о состоянии прибора и его датчиков, а также посредством взаимодействия с сервером.

Приложение на все запросы пользователя сначала пытается отправить запрос на сервер. При этом пользователь видит loadMask с надписью: “Загрузка данных...”. В случае получения от сервера ответа все данные записываются в локальную БД и обновляются соответствующие интерфейсные элементы.

В случае, если сеть не доступна или пользователь выбрал опцию “Использовать только смс-сообщения” на экране [“Авторизация”](#_1hmsyys) - приложение отправляет только смс-сообщение прибору.

Структура локальной БД соответствует структуре БД на сервере ([Структура базы данных](#m0d604j30dx4)).

Работа приложения посредством отправления/получения смс-сообщений различается для платформ Android и iOS.

1.1 Работа приложения. Платформа Android.

Платформа Android позволяет (при разрешении пользователя) отправлять/считывать смс-сообщения. Поэтому при получении смс-сообщения, отправленного с прибора - происходит считывание и расшифровка смс сообщения, полученная информация записывается в локальную БД, затем обновляется соответствующие блоки интерфейса (если они активны) приложения. Описание всех экранов в данном ТЗ соответствует работе приложения для платформы Android.

1.2 Работа приложения. Платформа iOS.

Платформа iOS не позволяет отправлять смс-сообщения без участия пользователя и считывать их. Поэтому, при взаимодействии пользователя с интерфейсом приложения (например, при нажатии кнопок) формируется смс-сообщение для прибора - пользователь должен отправить его с помощью средств отправки смс встроенных в ОС. После отправки СМС-сообщения пользователю показывается диалоговое окно с расшифровкой значений, которые могут быть получены от прибора в смс-сообщении. При получении смс-сообщения, отправленного с прибора - пользователь сможет только прочитать его сам. В локальную БД записываются только изменения, сделанные пользователем при взаимодействии с интерфейсом.

Логика получения и хранения данных

В мобильном приложении будет использовано два способа получения и отправки данных на сервер.

1. Первый способ - REST API, отправка данных на сервер посредством POST, PUT, GET, DELETE запросов.
2. Второй способ - открытие веб сокета (WebSocket), через который сервер отдает данные в приложение, а приложение прослушивает получение этих данных и при получении обрабатывает их.

В таблице [Список API запросов](#nnxuohifisqa) для каждого запроса данных указано, каким способом он будет производиться.

При получении сообщения о тревоге приложение должно обеспечить показ пользователя диалогового окна с сообщением о тревоге. В случае если это приложение “ИПРО-7”, приложение также должно обеспечить переход на вкладку [“Видеодомофон”](#rdohz9kkxqcc).

Все полученные данные должны сохраняться в локальной базе данных приложения.

Для поддержания актуальности информации о состоянии прибора предусмотрена опция - “обновлять по времени” ([Экран “Основные параметры”](#_sqyw64)) - пользователь может включить эту опцию и задать период времени, по истечении которого приложение будет посылать запрос о состоянии прибора на сервер.

Обработка сообщений об ошибках, приходящих с сервера

При любом взаимодействии приложения с сервером в режиме online могут происходить следующие ошибки, которые приложение должно корректно обрабатывать:

1. Сервер недоступен;
2. Время ожидания ответа превышено
3. Ошибка устройства

В случае недоступности сервера (сервер недоступен, нет доступа к сети) пользователю показывается диалоговое окно с сообщением об ошибке и предложением отправить запрос при помощи смс-сообщения. Диалоговое окно содержит две кнопки: “Ок” и “Отмена”. В случае, если пользователь выбирает кнопку “Ок” - приложение отправляет смс-сообщение прибору с соответствующей командой. В случае, если пользователь выбирает кнопку “Отмена”, диалоговое окно закрывается, никаких действий больше не происходит.

Так же будут обрабатываться прочие ошибки, не вошедшие в список.

При обработке ошибки пользователю будет отображаться модальное окно с сообщением об ошибке, полученной от сервера. В случае, если для ошибки с сервера не пришел текст сообщения, то будет отображаться стандартное сообщение об ошибке “Во время работы приложения произошла ошибка”.

Обработка сообщений об ошибках при работе с SMS-сообщениями

При любом взаимодействии приложения с смс-сообщениями могут происходить следующие ошибки, которые приложение должно корректно обрабатывать:

1. Ошибка при отправке смс-сообщения;
2. Услуга отправки смс-сообщений недоступна.
3. Отсутствует PDU - [Protocol Description Unit](https://ru.wikipedia.org/w/index.php?title=Protocol_Description_Unit&action=edit&redlink=1) (протокол отправки сообщений);
4. Ошибка - если пользователь запретил приложению отправлять/читать смс-сообщения.

Также будут обрабатываться прочие ошибки, не вошедшие в список.

При обработке ошибки пользователю будет отображаться модальное окно с сообщением об ошибке. Для ошибок, не вошедших в список выше и не описанных в данном ТЗ, будет отображаться стандартное сообщение об ошибке “Во время работы приложения произошла ошибка”.

Общая информация

Некоторые общие и платформо-зависимые элементы приложения.

Строка состояния

В верхней части приложения отображаются:

* Системная строка состояния (status bar);
* Панель навигации (navigation bar).

Панель навигации

Панель навигации может содержать:

* Текст с названием экрана;
* Кнопку “Назад” (слева);
* Другие кнопки (справа).

Текст с названием экрана может целиком не помещаться в отведенное для него место в панели навигации. В таком случае он принудительно обрезается и в конце добавляется многоточие.

При нажатии на кнопку “Назад” в панели навигации пользователь возвращается на предыдущий экран, находящийся в том же состоянии, если не указано иное.

Модальные и диалоговые окна

Для отображения различных сообщений пользователю в приложении используются стандартные для платформ iOS и Android модальные и диалоговые окна.

* Модальное окно используется, если пользователю необходимо отобразить некоторое сообщение (например, сообщение об ошибке). Оно содержит заголовок, текст сообщения и кнопку “OK”. При нажатии на кнопку “OK” модальное окно закрывается и никаких дополнительных действий не выполняется. Текст на кнопке может быть изменен.
* Диалоговое окно используется, если пользователь должен подтвердить некоторое действие перед его выполнением (например, сброс настроек) и обычно содержит две кнопки: “Отмена” и “OK”. При нажатии на кнопку “Отмена” диалоговое окно закрывается и никаких дополнительных действий не выполняется. При нажатии на кнопку “OK” выполняется действие пользователя. Текст на кнопках может быть изменен.

Навигация в приложении

Приложение имеет меню, которое открывается по нажатию на кнопку “Меню” расположенную слева в навигационной панели. Кнопка меню использует стандартную для этих целей иконку-гамбургер (≡).

Также меню открывается движением пальца слева-направо начиная с левого края экрана - оттягивая таким образом меню – меню будет открываться в этих случаях только с тех экранов, на которых есть кнопка “Меню” в навигационной панели слева.

Экран меню представляет собой панель, которая появляется на экране слева направо, не доходя до его правого края. Текущий экран при этом сдвигается вправо (iOS) или перекрывается панелью меню (Android). Область в правой части экрана, занятая текущим экраном, затеняется (Android). При нажатии по ней меню закрывается, текущий экран не меняется.

На экране меню расположена панель переключения между доступными пользователю приборами, в которой отображается название активного прибора, при нажатии на название пользователь переходит на экран [“Отображение и выбор используемого прибора”](#_2lwamvv). Ниже расположен список пунктов меню. При выборе одного из пунктов списка, меню закрывается, а текущий экран заменяется выбранным разделом (если не указано иное).

Поведение приложения при нажатии кнопки “Назад” на устройствах с ОС Android

Приоритет закрытия элементов в момент, когда пользователь нажал кнопку “Назад”:

1. Если открыта клавиатура для ввода данных ­ клавиатура скроется.
2. Если открыто модальное окно, нажатие кнопки “Назад” закрывает данное окно.
3. Если на экране находится диалоговое окно с кнопками выбора “Отмена” и “OK”, нажатие кнопки “Назад” закрывает диалоговое окно с автовыбором “Отмена”.
4. Если на экране находится диалоговое окно с одной̆ кнопкой̆, нажатие кнопки “Back“ закрывает диалоговое окно с автонажатием на данную кнопку.
5. Нажатие кнопки “Назад” на любом экране у которого есть наверху кнопка “Назад” выполняет переход на предыдущий экран. В случае, если на этом экране есть несохраненные данные - пользователю показывается диалоговое окно с сообщением, о наличии несохраненных данных и двумя кнопками “Сохранить” и “Отмена”.
6. Если пользователь находится на экране меню, то по нажатию на кнопку “Назад” меню закрывается.
7. Если пользователь находится на экране, на котором нет кнопки “Назад” и нет элементов, которые можно закрыть, то по нажатию на кнопку “Назад” приложение закрывается.

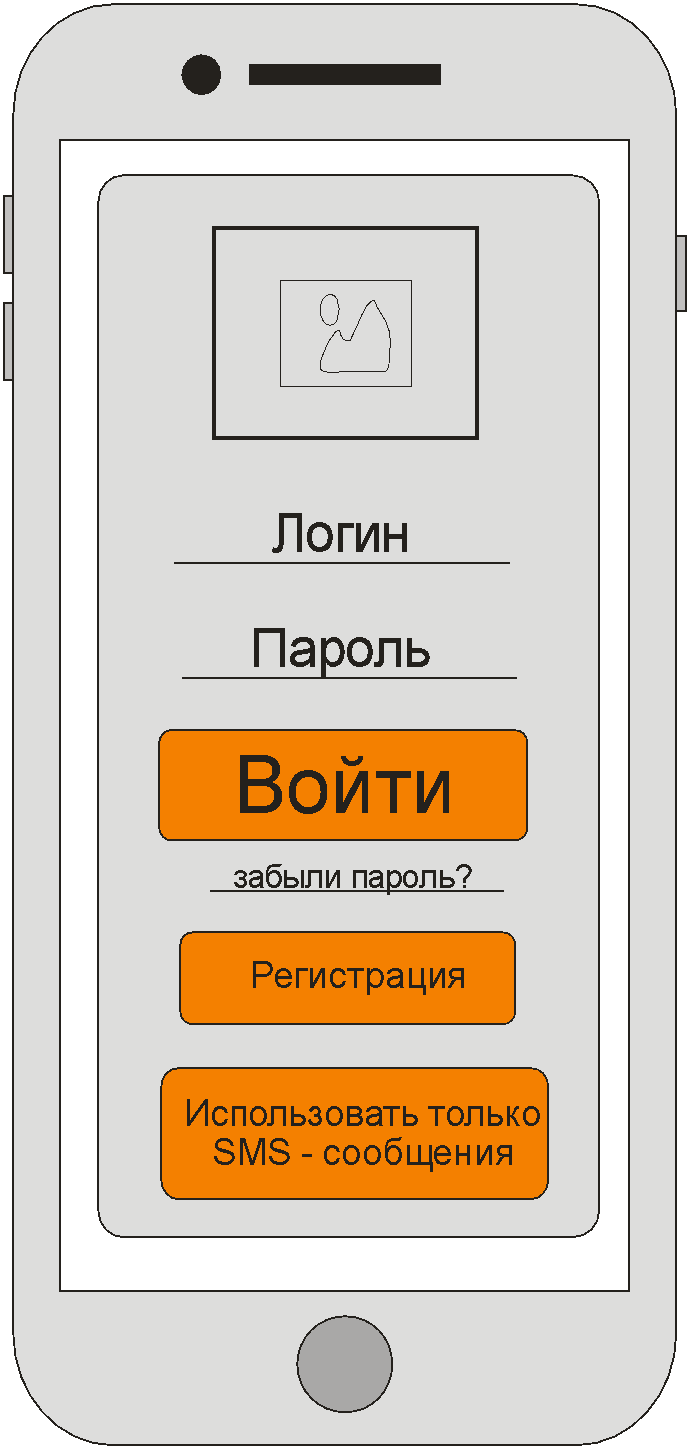
* 1. Описание работы приложения “ИПРО-7”

Запуск приложения

При запуске приложения проверяется - есть ли запись о пользователе в локальной БД. В случае, если запись о пользователе в локальной БД есть - пользователю показывается экран [“Ввод пароля”](#_4f1mdlm) или [Основной экран (Главная)](#_2u6wntf) - в зависимости от настроек, сохраненных в БД ([“Настройка”](#_2dlolyb)). В противном случае - пользователь переходит на экран [“Авторизация”](#_1hmsyys).

**Экран “Авторизация”**

На экране “Авторизация” (Рисунок 1) должны быть организованы следующие функциональные требования:



Этот запрос отправляет прибор.

Модель запроса: **Auth**. Поля:

* login - IMEI номер прибора (15 символов)
* password - пароль пользователя на приборе

Модель ответа: **AuthResult**. Поля:

* accessToken - токен авторизации
* expiresIn - дата и время истечения срока действия токена

Рисунок 1 - Экран "Авторизация"

1. Пользователь должен иметь возможность вводить информацию, необходимую для идентификации пользователя на сервере: логин и пароль пользователя.
2. Приложение должно обеспечивать валидацию введенной пользователем информации после нажатия пользователем на кнопку “Войти”. А также сообщать пользователю о найденных проблемах. Валидация включает в себя:

* проверка ввода пользователем логина и пароля.

1. Приложение должно обеспечить отправку данных, введенных пользователем на сервер с помощью запроса ([Авторизация](#_2jh5peh)), сохранение данных, введенных пользователем в локальную БД. А также загрузку списка приборов ([Получить список приборов](#_3fg1ce0)), доступных пользователю.
2. В случае, если у пользователя есть зарегистрированный на сервере прибор приложение должно обеспечить загрузку данных для первого прибора в списке ([Запросить состояние прибора](#_1vsw3ci)) и переход на [Главный экран](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.dj8cmyonea00).
3. В случае, если список приборов, зарегистрированных у пользователя, пуст - приложение должно обеспечить переход на экран [“Регистрация прибора”](#_1v1yuxt).
4. В случае, если сервер не доступен - приложение должно обеспечить работу приложения путем отправки смс-сообщений прибору. Подробнее об этом описано в разделе [Принципы работы приложения](#_1y810tw).
5. Приложение должно обеспечить переход на экран ["Регистрация пользователя”](#_3jd0qos) при нажатии пользователя на кнопку “Регистрация”.
6. Приложение должно переходить в режим смс-общения с прибором ([Принципы работы приложения](#_1y810tw)) при нажатии пользователя на кнопку “Использовать только смс-сообщения”.

**Экран “Регистрация пользователя”**

На экране “Регистрация пользователя” (Рисунок 2) должны быть организованы следующие функциональные требования:

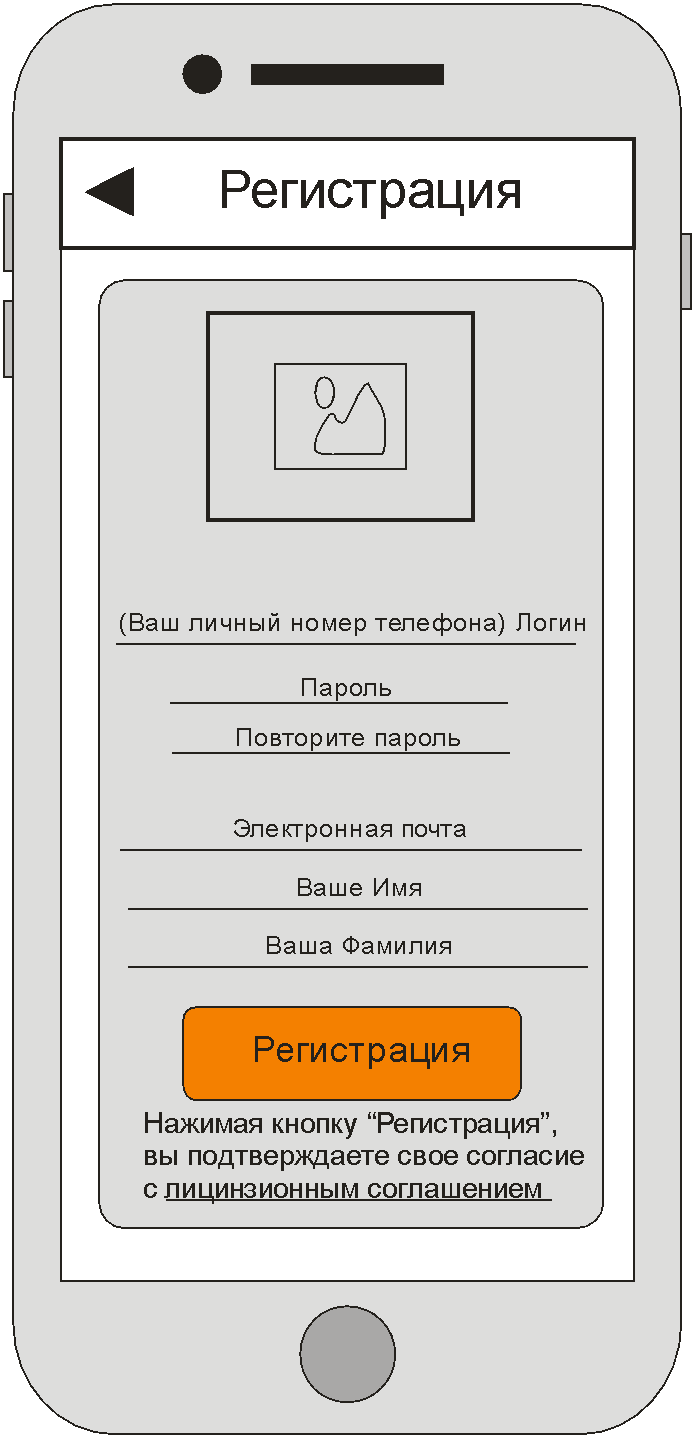
Модель запроса: **CreateUser**. Поля:

Рисунок - экран "Регистрация пользователя"

* username - логин (им является номер телефона), обязательное поле
* password - пароль, обязательное поле
* firstname - имя, необязательное поле
* lastname - фамилия, необязательное поле
* email - электронная почта, необязательное поле
* regionId - идентификатор региона, обязательное поле
* isAgree - 0 - не согласен, 1 - согласен с условиями и положениями (Terms & Conditions), обязательное поле

Модель ответа: **CreateUserResult**. Поля:

* isCreated - 0 - пользователь не создан, 1 - пользователь создан

1. Пользователь должен иметь возможность вводить информацию для регистрации на сервере: номер телефон, пароль (и его подтверждение) - обязательные для заполнения поля. Необязательные, но при их отсутствии (при нажатии кнопки зарегистрировать) должно выводиться предупреждение, например: "Укажите ваше имя, фамилию и адрес электронной почты, без этих данных восстановить учетную запись в случае утраты пароля не получится". Необязательный пункт - регион. Под формой ввода данных разместить текст «Я даю согласие на обработку персональных данных» и чекбокс.
2. Приложение должно обеспечивать валидацию введенной пользователем информации после нажатия пользователем на кнопку “Регистрация”. А также сообщать пользователю о найденных проблемах. Валидация включает в себя:

* проверка ввода пользователем номера телефона (количество цифр в поле ввода не должно быть меньше 6) - в поле телефона можно вводить только цифры;
* Значения полей “Пароль” и “Подтвердите пароль” должны быть одинаковыми. (количество символов не меньше 4);

1. Приложение должно обеспечить регистрацию пользователя на сервере ([Регистрация пользователя](#_2981zbj)) и сохранение введенных пользователем данных в БД ([Таблица 1. Users](#_2vor4mt));
2. Приложение должно обеспечить переход приложения на экран [“Авторизация”](#_1hmsyys), в котором предзаполняются поля введенными пользователем логином и паролем, в случае успешного прохождения данными валидации.
3. Приложение должно обеспечивать переход приложения на веб-страницу с лицензионным соглашением, в случае если пользователь нажимает на кнопку “Лицензионное соглашение”.

**Экран “Регистрация прибора”**

На экране “Регистрация прибора” (Рисунок 3) должны быть организованы следующие функциональные требования:

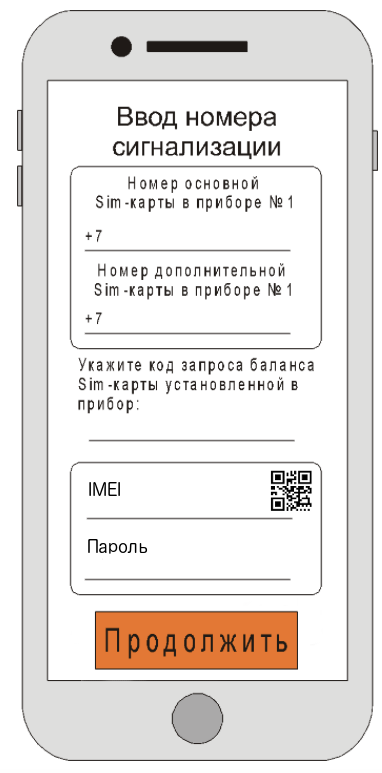
Модель запроса: **CreateDevice**. Поля:

Рисунок - экране “Регистрация прибора”

* number - IMEI номер прибора (15 символов)
* name - название прибора
* mainSim - номер основной Sim-карты в приборе
* additionalSim - номер дополнительной Sim-карты в приборе
* balanceCode - код запроса баланса Sim-карты, установленной в прибор
* type - тип прибора, поле типа **DeviceType**

Модель ответа: **CreateDeviceResult**. Поля:

* isCreated - булевый флаг, прибор создан/не создан

**DeviceType.** Перечисление типа:

1 – ИПРО-7

1. Пользователь должен иметь возможность вводить информацию для идентификации прибора: имя прибора, номера основной и дополнительной Sim карты в приборе, код запроса баланса, а также, в случае если приложение находится в режиме онлайн (подробнее об этом в [Принципы работы приложения “ИПРО-7”](#_1y810tw)) - IMEI прибора и пароль доступа к прибору.
2. Пользователь должен иметь возможность отсканировать IMEI прибора - для этого используется кнопка отсканировать QR-код, по нажатию на которую приложение переходит в окно сканера, а после завершения сканирования полученным кодом заполняется поле “IMEI”;
3. Приложение должно обеспечивать валидацию введенной пользователем информации после нажатия пользователем на кнопку “Продолжить”. А также сообщать пользователю о найденных проблемах. Валидация включает в себя:

* проверка ввода пользователем номера основной sim-карты в приборе;
* проверка ввода пользователем IMEI прибора (пользователь должен ввести 15 символов) и пароля доступа к прибору (в случае если приложение находится в режиме онлайн);
* проверка введенных номеров телефонов - состоят только из цифр, количество цифр должно быть больше 6.

1. Приложение должно обеспечивать сохранение данных, введенных пользователем в локальную БД в таблицу ([Таблица 2. Devices](#_1au1eum)).
2. Приложение должно обеспечивать отправку данных, введенных пользователем на сервер при помощи запроса ([Добавить прибор](#_38czs75)) или смс-сообщения ([Запись телефонного номера в телефонную книгу прибора](#_z337ya) ).
3. Приложение должно обеспечивать переход приложения на экран “[Основной экран (Главная)](#_2u6wntf)”.

**Экран “Ввод пароля”**

По умолчанию этот экране не показывается в приложении. Пользователь может включить вход в приложение по паролю на экране [“Настройка”](#_2dlolyb). На экране “Ввод пароля” (Рисунок 4) должны быть организованы следующие функциональные требования:

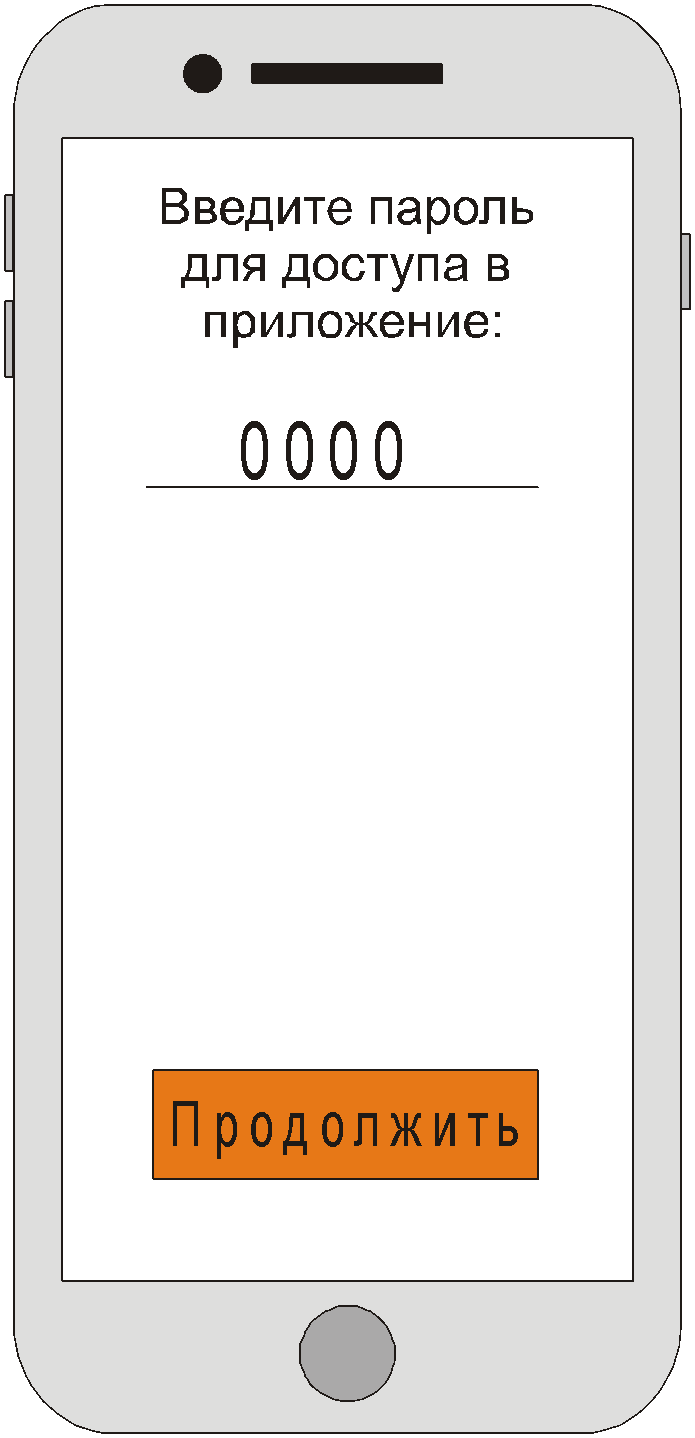


Рисунок - экран "Ввод пароля"

1. Пользователь должен иметь возможность ввести пароль для доступа в приложение, заданный на экране [“Настройка”](#_2dlolyb).
2. Приложение должно обеспечивать сравнение введенного пользователем пароля с паролем, хранящимся в БД. В случае полного совпадения - должен быть осуществлен переход на экран “[Основной экран (Главная)](#_2u6wntf)”, в противном случае - приложение должно показать пользователю сообщение о неверном вводе пароля.

**Основной экран (Главная)**

На экране “Основной экран” (Рисунок 5) должны быть организованы следующие функциональные требования:

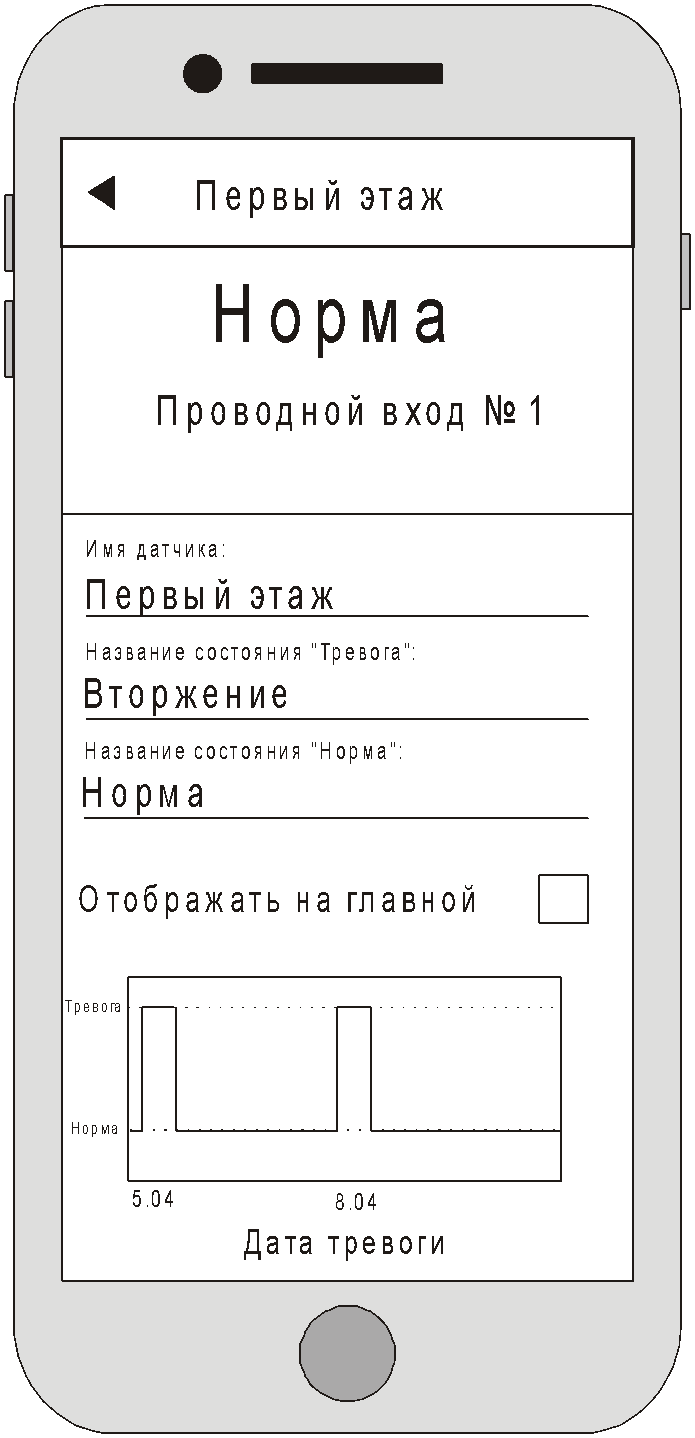
1. Приложение должно обеспечивать отображение состояния прибора, датчиков и выходов в понятной пользователю форме. Для этого используются плитки с информацией - каждая плитка содержит название датчика, название параметра и текущее значение параметра.

Рисунок - экран "Основной экран"

1. Приложение должно обеспечивать отображение только той информации, которую выбрал пользователь на экране [“Отобразить на главной”](#_46r0co2).
2. Приложение должно обеспечить отображение информации о состоянии баланса на основной сим карте, вставленной в прибор. Значение баланса берется из локальной БД.
3. Приложение должно обеспечить переход на экран [“Отобразить на главной”](#_46r0co2) при нажатии пользователем на кнопку “+”.
4. Приложение должно обеспечивать постановку (снятие) прибора с охраны при нажатии пользователем на кнопку “Поставить на охрану” (“Снять с охраны”). Работа приложения в этом случае должна быть обеспечена или запросом к серверу ([Поставить/снять все зоны с охраны](#_243i4a2)) или отправлением смс-сообщения [Поставить на охрану](#_tyjcwt) ([Снять с охраны](#_3dy6vkm)) непосредственно на прибор (в случае если не удалось связаться с сервером или серверу не удалось связаться с прибором или приложение работает в режиме оффлайн - подробнее это описано в [Принципы работы приложения “ИПРО-7”](#_1y810tw)). Также приложение должно обеспечить информирование пользователя о состоянии прибора (снят с охраны или нет) путем изменения состояния соответствующей кнопки.
5. Приложение должно обеспечить проверку баланса основной сим карты прибора при нажатии пользователя на кнопку “Запросить баланс”. Работа приложения в этом случае должна быть обеспечена или запросом к серверу ([Запросить баланс](#_j8sehv)) или отправлением смс-сообщения [Запрос Баланса](#_3j2qqm3) непосредственно на прибор (в случае если не удалось связаться с сервером или серверу не удалось связаться с прибором или приложение работает в режиме оффлайн - подробнее это описано в [Принципы работы приложения “ИПРО-7”](#_1y810tw)).
6. Приложение должно обеспечить сохранение в локальную БД данных о постановке (снятии) прибора на охрану и о состоянии баланса основной сим карты в приборе.
7. Приложение должно обеспечить переход на соответствующие экраны с детальной информацией, при нажатии пользователь на плитку с информацией о состоянии прибора, его входов и выходов

**Экран “Проводной вход”**

На экране “Проводной вход” (Рисунок 6) приложение должно обеспечить ввод параметров отображения на [Основном экране](#_2u6wntf) информации о проводном входе прибора. На экране должны быть организованы следующие функциональные требования:



Включить/выключить вход

Модель запроса: **InputOn**. Поля:

* id - идентификатор прибора
* id - номер входа (1, 2, 3 и т.д.)
* inputType - тип входа, 0 - беспроводной вход, 1 - проводной вход
* isOn - 0 - выключить вход, 1 - включить вход, 2 - запросить состояние входа (включен/выключен)

Модель ответа: **InputOnResult**. Поля:

* + - * + isOn - 0 - вход выключен, 1 - вход включен

Cохранить настройки входа

Рисунок - экран “Проводной вход”

Модель запроса: **InputSettings.** Поля:

* id - идентификатор прибора
* name - название входа
* alarm - название состояния “Тревога”
* norma - название состояния “Норма”
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

Модель ответа: **InputSettingsResult.** Поля:

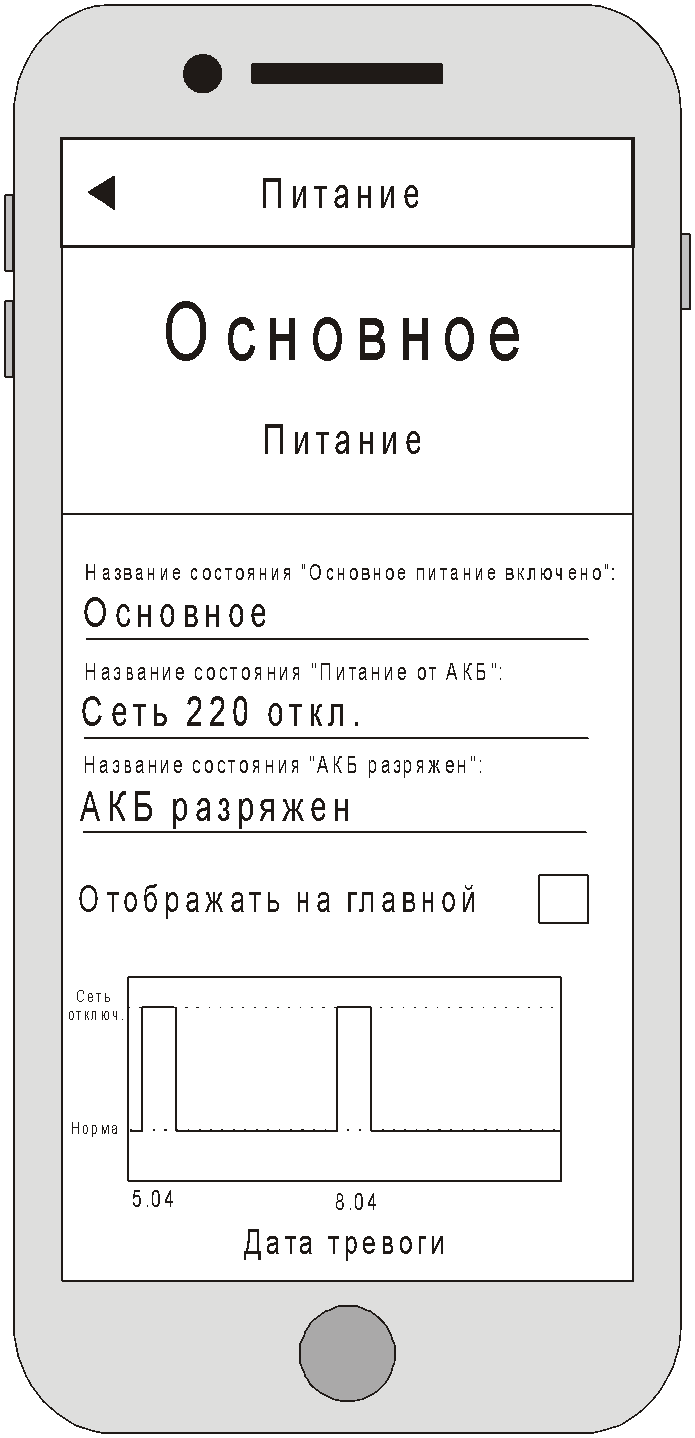
* isSaved - 0 - настройки не сохранены, 1 - настройки сохранены

1. Приложение должно обеспечивать отображение состояния проводного входа прибора, его пользовательское название и название в приборе, а также статистики состояния прибора (за период, сохраненный в локальной БД).
2. Пользователь должен иметь возможность изменять пользовательское название проводного входа (по умолчанию это название совпадает с названием входа на приборе).
3. Пользователь должен иметь возможность изменять названия для состояния “Тревога” и “Норма” (по умолчанию эти названия: “Тревога” и “Норма”).
4. Пользователь должен иметь возможность выбрать: отображать информацию о данном проводном входе на [Основной экран](#_2u6wntf) или нет - для этого используется чекбокс “Отображать на главной”.
5. Пользователь должен иметь возможность аварийно отключить вход - для этого используется кнопка “Аварийное отключение входа”. При нажатии на эту кнопку серверу отправляется запрос [Включить/выключить вход](#7vnjv74vhey0), в случае недоступности сервера - отправляется смс-сообщение [Включение/отключение проводного входа](#wpo7uul74ef8).
6. Приложение должно обеспечивать сохранение изменений, сделанных пользователем, после того как пользователь покидает экран путем нажатия на кнопку “Назад” или кнопку “Back” (для Android). Сохранение должно происходить следующим образом:

* В случае, если какое-то поле ввода (имя входа, название состояния “Тревога” или “название состояния “Норма”) пустые - используются значения по умолчанию;
* Приложение должно сравнивать значения, введенные пользователем в поля ввода и значения, сохраненные в локальной БД - в случае если эти значения не совпадают, на сервер (в случае работы в режиме online) отправляется запрос ([Cохранить настройки входа](#_261ztfg)), после успешного ответа сервера или если сервер не доступен - изменения пользователя должны быть записаны в локальной БД ([Таблица 3. Inputs](#_3utoxif)).

**Экран “Питание”**

На экране “Питание” (Рисунок 7) приложение должно обеспечить ввод параметров отображения на [Основном экране](#_2u6wntf) информации о питании прибора. На экране должны быть организованы следующие функциональные требования:

Cохранить настройки питания

Модель запроса: **PowerSettings.** Поля:

* id - идентификатор прибора
* powerName - название состояния “Основное питание включено”
* powerAkb - название состояния “Питание от АКБ”
* powerOff - название состояния “АКБ разряжен”
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

Модель ответа: **PowerSettingsResult.** Поля:

* isSaved - 0 - настройки не сохранены, 1 - настройки сохранены

1. Приложение должно обеспечивать отображение состояния питания прибора, а также статистику состояния питания прибора (за период, сохраненный в локальной БД).

Рисунок - экран “Питание”

1. Пользователь должен иметь возможность изменять названия для состояния “Основное питание включено”, “Питание от АКБ” и “АКБ разряжен” (по умолчанию эти названия: “Основное” и “Сеть 220 откл” и “АКБ разряжен”).
2. Пользователь должен иметь возможность выбрать: отображать информацию о питании прибора на [Основной экран](#_2u6wntf) или нет - для этого используется чекбокс “Отображать на главной”.
3. Приложение должно обеспечивать сохранение изменений, сделанных пользователем, после того как пользователь покидает экран путем нажатия на кнопку “Назад” или кнопку “Back” (для Android). Сохранение должно происходить следующим образом:

* В случае, если какое-то поле ввода (названия для состояния “Основное питание включено”, “Питание от АКБ” и “АКБ разряжен””) пустые - используются значения по умолчанию;
* Приложение должно сравнивать значения, введенные пользователем в поля ввода и значения, сохраненные в локальной БД - в случае если эти значения не совпадают, на сервер (в случае работы в режиме online) отправляется запрос ([Cохранить настройки питания](#_3qwpj7n)), после успешного ответа сервера или если сервер не доступен - изменения пользователя должны быть записаны в локальной БД ([Таблица 2. Devices](#_1au1eum)).

**Экран “Беспроводной вход”**

На экране “Беспроводной вход” (Рисунок 8) приложение должно обеспечить ввод параметров отображения на [Основном экране](#_2u6wntf) информации о беспроводном входе прибора. На экране должны быть организованы следующие функциональные требования:

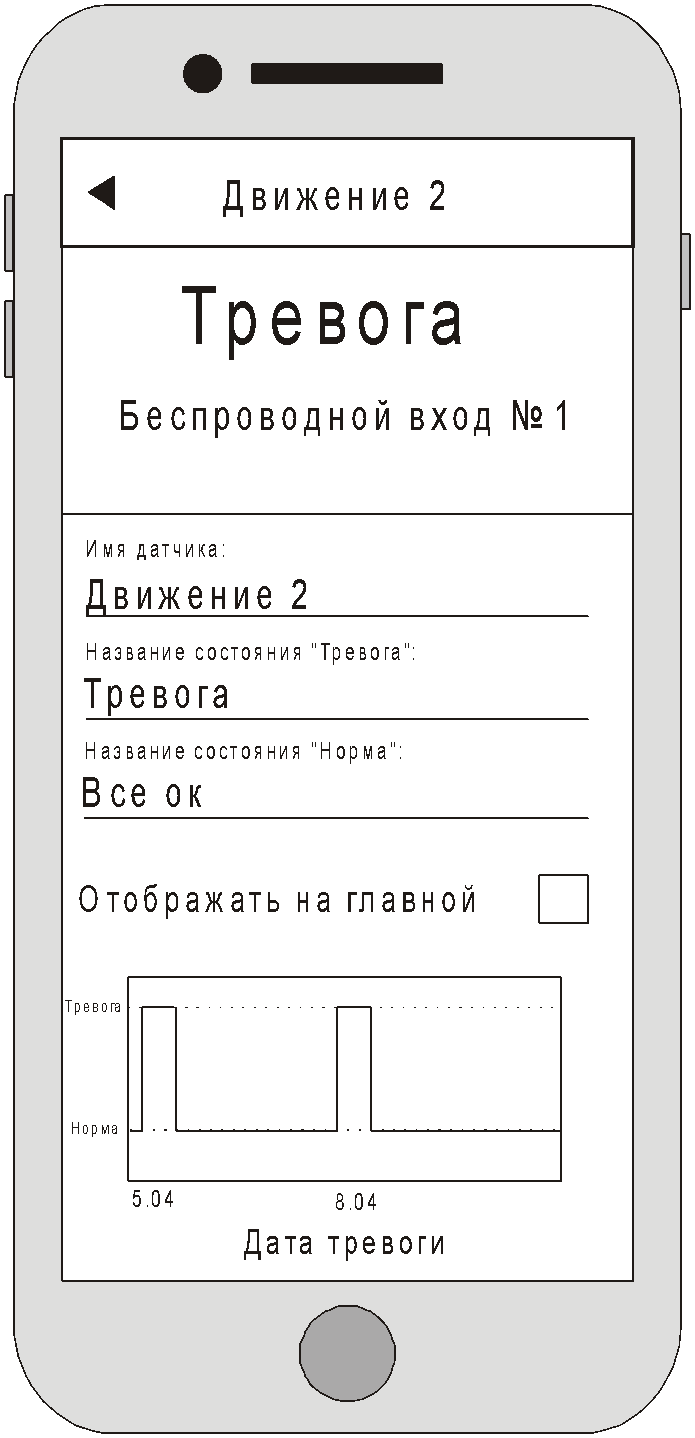
Включить/выключить вход

Рисунок - экран “Беспроводной вход”

Модель запроса: **InputOn**. Поля:

* id - идентификатор прибора

id - номер входа (1, 2, 3 и т.д.)

* inputType - тип входа, 0 - беспроводной вход, 1 - проводной вход
* isOn - 0 - выключить вход, 1 - включить вход, 2 - запросить состояние входа (включен/выключен)

Модель ответа: **InputOnResult**. Поля:

isOn - 0 - вход выключен, 1 - вход включен

Cохранить настройки входа

Модель запроса: **InputSettings.** Поля:

* id - идентификатор прибора
* name - название входа
* alarm - название состояния “Тревога”
* norma - название состояния “Норма”
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

Модель ответа: **InputSettingsResult.** Поля:

* isSaved - 0 - настройки не сохранены, 1 - настройки сохранены

1. Приложение должно обеспечивать отображение состояния беспроводного входа прибора, его пользовательское название и название в приборе, а также статистики состояния прибора (за период, сохраненный в локальной БД).
2. Пользователь должен иметь возможность изменять пользовательское название беспроводного входа (по умолчанию это название совпадает с названием входа на приборе).
3. Пользователь должен иметь возможность изменять названия для состояния “Тревога” и “Норма” (по умолчанию эти названия: “Тревога” и “Норма”).
4. Пользователь должен иметь возможность выбрать: отображать информацию о данном беспроводном входе на [Основной экран](#_2u6wntf) или нет - для этого используется чекбокс “Отображать на главной”.
5. Пользователь должен иметь возможность аварийно отключить вход - для этого используется кнопка “Аварийное отключение входа”. При нажатии на эту кнопку серверу отправляется запрос [Включить/выключить вход](#7vnjv74vhey0), в случае недоступности сервера - отправляется смс-сообщение [Включение/отключение проводного входа](#wpo7uul74ef8).
6. Приложение должно обеспечивать сохранение изменений, сделанных пользователем, после того как пользователь покидает экран путем нажатия на кнопку “Назад” или кнопку “Back” (для Android). Сохранение должно происходить следующим образом:

* В случае, если какое-то поле ввода (имя входа, название состояния “Тревога” или название состояния “Норма”) пустое - используется значение по умолчанию;
* Приложение должно сравнивать значения, введенные пользователем в поля ввода и значения, сохраненные в локальной БД - в случае если эти значения не совпадают, на сервер (в случае работы в режиме online) отправляется запрос ([Cохранить настройки входа](#_261ztfg)), после успешного ответа сервера или если сервер не доступен - изменения пользователя должны быть записаны в локальной БД ([Таблица 3. Inputs](#_3utoxif)).

**Экран “Выход”**

На экране “Выход” (Рисунок 9) приложение должно обеспечить ввод параметров отображения на [Основном экране](#_2u6wntf) информации о выходе прибора. На экране должны быть организованы следующие функциональные требования:

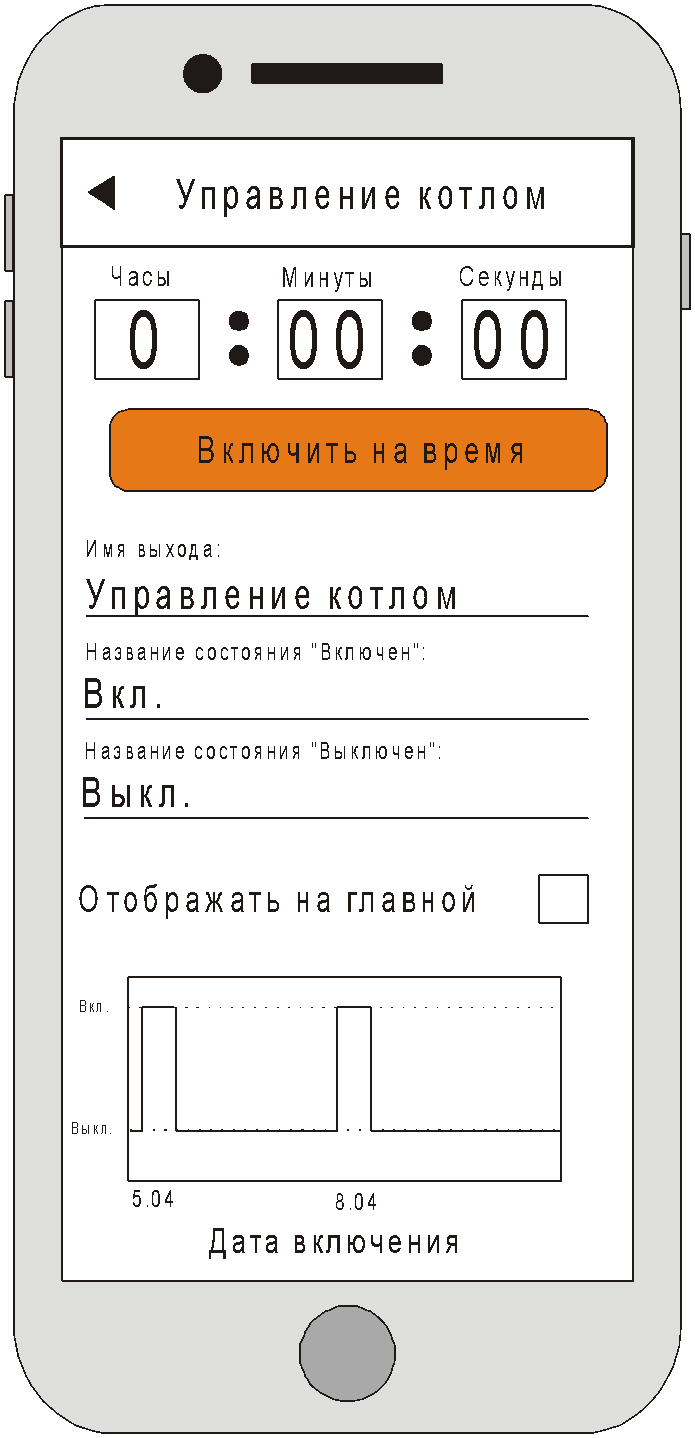


Включить/выключить выход

Модель запроса: **OutputOn** Поля:

* id - идентификатор прибора
* id - номер выхода (1, 2, 3 и т.д.)
* isOn - 0 - выключить выход, 1 - включить выход
* isOnTime - 0 - не зависит от времени, 1 - включить выход на время
* time - время на которое нужно включить выход в секундах

Модель ответа: **OutputOnResult**. Поля:

*  isOn - 0 - выход выключен, 1 - выход включен, 2 - запросить состояние (включен/выключен)

Cохранить настройки выхода

Модель запроса: **OutputSettings.** Поля:

* id - идентификатор прибора
* name - название выхода
* onState - название состояния “Включен”
* offState - название состояния “Выключен”
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

Модель ответа: **OutputSettingsResult.** Поля:

* isSaved - 0 - настройки не сохранены, 1 - настройки сохранены

1. Приложение должно обеспечивать отображение пользователю текущего состояния выхода, его пользовательское название и название в приборе, а также статистики состояния выхода прибора (за период, сохраненный в локальной БД).

Рисунок - экран “Выход”

1. Пользователь должен иметь возможность изменять пользовательское название выхода (по умолчанию это название совпадает с названием выхода на приборе).
2. Пользователь должен иметь возможность изменять названия для состояния “Включен” и “Выключен” (по умолчанию эти названия: “Включен” и “Выключен”).
3. Пользователь должен иметь возможность задать промежуток, в течении которого выход должен быть включен - для этого используется диалоговое окно для задания промежутка времени, в котором содержатся три слайдера (Рисунок 1.10) - для задания интервала в часах (от 0 до 9), в минутах (от 0 до 59) и секундах (от 0 до 59).
4. Пользователь должен иметь возможность включить, выключить или включить выход на определенное время - для этого в приложении должны быть предусмотрены соответствующие интерфейсные элементы (кнопки для включения/выключения выхода, пикер для задания временного промежутка (Рисунок 1.10)). Включение/выключение выхода должно производиться следующим образом:

* на сервер отправляется соответствующий запрос ([Включить/выключить выход](#_2eclud0))
* в случае если связь с сервером по каким-то причинам не доступна - на прибор отправляется смс-сообщение [Управление выходами](#_1t3h5sf) (подробнее об отправке смс в [Принципы работы приложения “ИПРО-7”](#_1y810tw).
* данные о запросе записываются в локальную БД. Состояние прибора обновляется и так же записывается в локальную БД.

1. Пользователь должен иметь возможность выбрать: отображать информацию о данном выходе на [Основном экране](#_2u6wntf) или нет - для этого используется чекбокс “Отображать на главной”.
2. Приложение должно обеспечивать сохранение изменений сделанных пользователем, после того как пользователь покидает экран путем нажатия на кнопку “Назад” или кнопку “Back” (для Android). Сохранение должно происходить следующим образом:

* В случае, если какое-то поле для выхода (имя выхода, название состояния “Включен” или название состояния “Выключен”) пустое - используется значение по умолчанию;
* Приложение должно сравнивать значения, введенные пользователем в поля ввода и значения, сохраненные в локальной БД - в случае если эти значения не совпадают, на сервер (в случае работы в режиме online) отправляется запрос ([Cохранить настройки выхода](#_l7a3n9)), после успешного ответа сервера или если сервер не доступен - изменения пользователя должны быть записаны в локальной БД ([Таблица 4. Outputs](#_29yz7q8)).

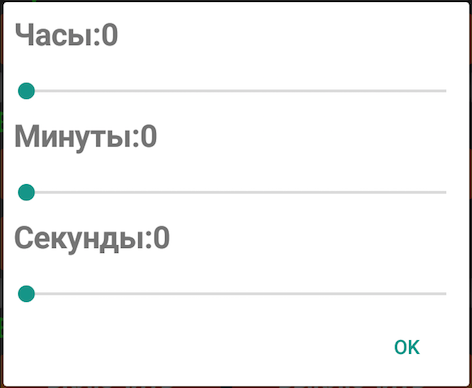


Рисунок - Диалоговое окно для задания интервала времени

**Экран “Датчик температуры”**

На экране “Датчик температуры” (Рисунок 11) приложение должно обеспечить ввод параметров отображения на [Основном экране](#_2u6wntf) информации о датчике температуры прибора. На экране должны быть организованы следующие функциональные требования:

1. Приложение должно обеспечивать отображение пользователю последнюю сохраненную в локальной БД температуру на термодатчике, его пользовательское название и название в приборе.
2. Пользователь должен иметь возможность изменять пользовательское название термодатчика (по умолчанию это название “Датчик температуры №” и номер термодатчика в приборе).

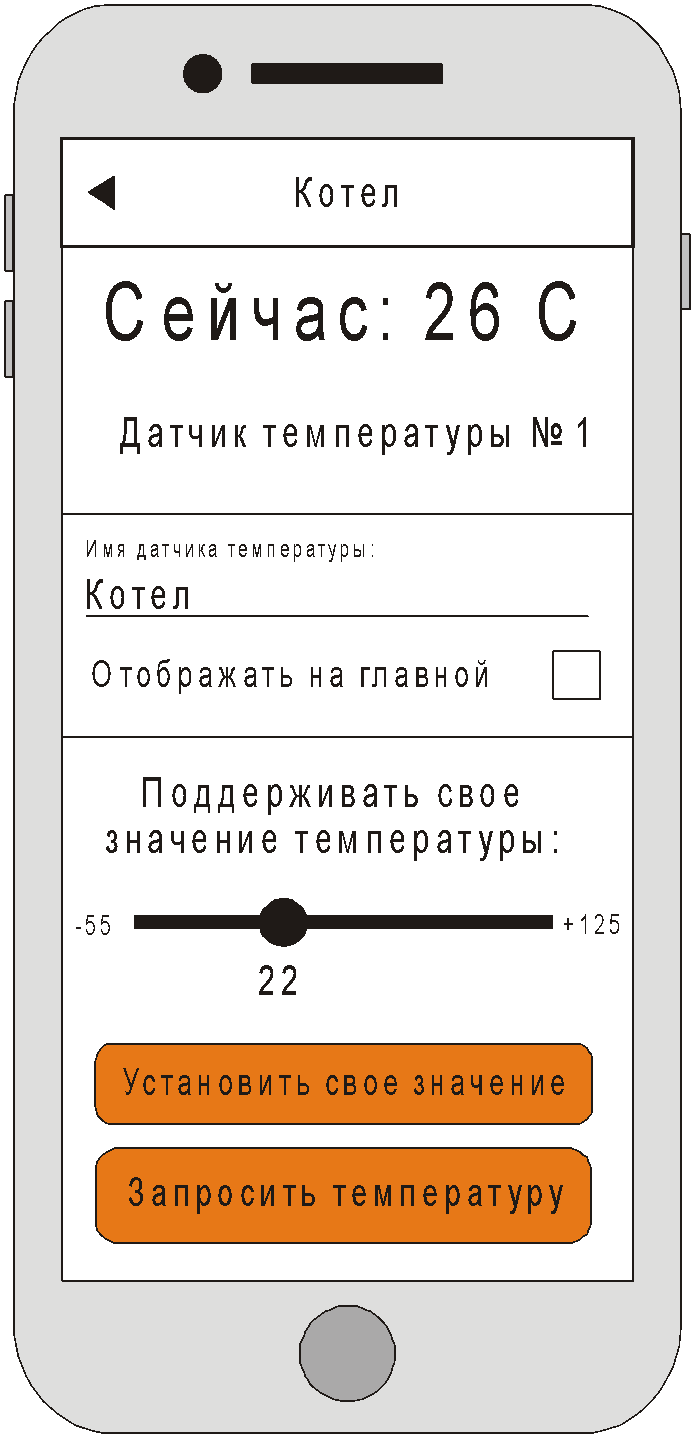
 Запросить температуру на термодатчике

Рисунок - экран “Датчик температуры”

Модель запроса: **GetTemperature**. Поля:

* id - идентификатор прибора
* id - номер термодатчика (1, 2, 3 и т.д.)

Модель ответа: **TemperatureSensorInfo**. Поля:

* degree - значение температуры в °С, которую фиксирует датчик

Cохранить настройки термодатчика

Модель запроса: **TemperatureSensorSettings.** Поля:

* id - идентификатор прибора
* name - название термодатчика
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

Модель ответа: **TemperatureSensortSettingsResult.** Поля:

* isSaved - 0 - настройки не сохранены, 1 - настройки сохранены

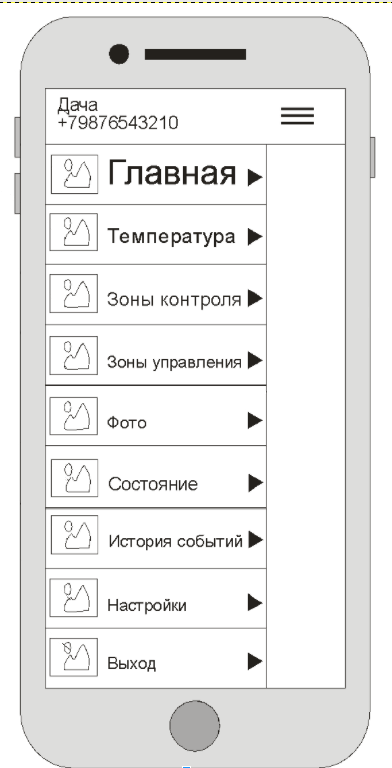
1. Пользователь должен иметь возможность запросить значение текущей температуры, регистрируемой датчиком путем нажатия на кнопку “Запросить температуру”. В случае если приложение работает в режиме online - на сервер посылается запрос ([Запросить температуру на термодатчике](#_thw4kt)), в случае если запрос не выполняется или в случае работы в режиме offline - прибору (по согласию пользователя) отсылается смс-сообщение [Запрос температуры](#_2s8eyo1) и после получения ответного смс (в случае, если приложение работает на платформе Android) - результат отображается в соответствующем поле. Подробнее о механизме работы приложения с сервером описано в [Принципы работы приложения “ИПРО-7](#_1y810tw).
2. Пользователь должен иметь возможность установить температуру, которую должен поддерживать прибор - для этого на экране предусмотрен слайдер (диапазон задаваемой температуры на слайдере от -55 до 125, шаг - 1) и кнопка “Установить свое значение”. При нажатии пользователя на эту кнопку на сервер отправляется запрос ([Установить температуру](#_3gnlt4p)) или отправляется смс сообщение [Управление датчиками температуры](#_4d34og8). Подробнее о механизме работы приложения с сервером описано в [Принципы работы приложения “ИПРО-7”](#_1y810tw).
3. Пользователь должен иметь возможность выбрать: отображать информацию о данном выходе на [Основном экране](#_2u6wntf) или нет - для этого используется чекбокс “Отображать на главной”.
4. Приложение должно обеспечивать сохранение изменений сделанных пользователем, после того как пользователь покидает экран путем нажатия на кнопку “Назад” или кнопку “Back” (для Android). Сохранение должно происходить следующим образом:

* В случае, если поле “имя термодатчика” пустое - используется значение по умолчанию;
* Приложение должно сравнивать значение, введенное пользователем в поле ввода и значение, сохраненные в локальной БД - в случае если эти значения не совпадают, на сервер (в случае работы в режиме online) отправляется запрос ([Сохранить настройки термодатчика](#_46ad4c2)), после успешного ответа сервера или если сервер не доступен - изменения пользователя должны быть записаны в локальной БД ([Таблица 7. TemperatureSensors](#_1o97atn)).

**Главное меню приложения**

Главное меню приложения (Рисунок 12) состоит из следующих элементов:

Меню должно содержать следующие пункты:

1. Название активного прибора, при нажатии на этот пункт происходит переход на экран [“Отображение и выбор используемого прибора”](#_2lwamvv).
2. “Главная” - открывает главное окно приложения ([Основной экран](#_2u6wntf)).
3. “Температура” - переходим в окно [“Температура”](#_206ipza).
4. “Зоны контроля” - переходим в окно [“Зоны контроля”](#_4k668n3).
5. Зоны управления - переходим в окно [“Зоны управления”](#_2zbgiuw).
6. Состояние - переход в окно [“Состояние”](#_1egqt2p).
7. История событий - переход в окно [“История событий”](#_3ygebqi).
8. Настройки - переход в окно [“Настройка”](#_2dlolyb).
9. Выход - происходит выход из приложения.

**Экран “Отобразить на главной”**

Рисунок - Главное меню приложения

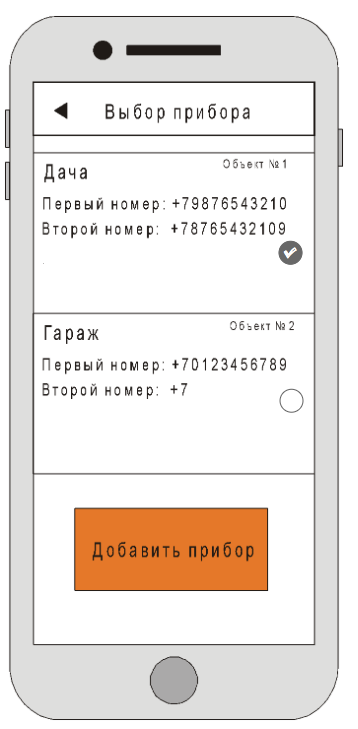
На экране “Отобразить на главной” (Рисунок 13) приложение должно предоставить пользователю возможность выбрать - какую информацию необходимо отобразить на [Основной экран (Главная)](#_2u6wntf). На экране должны быть организованы следующие функциональные требования:

1. Приложение должно обеспечить отображение всех доступных пользователю входов, выходов, датчиков температуры. Для каждого элемента необходимо выводить пользовательское название, название этого элемента в приборе.
2. Для каждого элемента должна быть предусмотрена возможность выключить или выключить отображение этого элемента на на главном экране ([Основной экран](#_2u6wntf)).

**Экран “Отображение и выбор используемого прибора”**

Рисунок - экран “Отобразить на главной”

На экране “Отображение и выбор используемого прибора” (Рисунок 14) приложение должно предоставить пользователю возможность выбрать - информацию о каком приборе необходимо отобразить на всех экранах приложения. На экране должны быть организованы следующие функциональные требования:



1. Приложение должно отображать список приборов, доступных пользователю. Каждый элемент списка должен отображать название прибора, порядковый номер (задается при добавлении прибора), индивидуальный номер прибора, номер телефона основной сим-карты, вставленной в прибор.
2. Приложение должно обеспечить возможность выбрать любой элемент списка для отображения информации о нем в приложении.
3. Приложение должно обеспечить удаление прибора из локальной БД и на сервере, при лонг тапе на соответствующую строчку в списке прибора и при получении согласия пользователя на это.
4. Приложение должно обеспечить переход на экран [“Редактировать прибор”](#_3l18frh) при выборе пользователем соответствующего элемента списка.

Рисунок - экран “Отображение и выбор используемого прибора”

1. Приложение должно обеспечить переход на экран [“Добавить прибор”](#_111kx3o) при нажатии пользователя на кнопку “Добавить прибор”.

**Экран “Добавить прибор”**

На экране “Добавить прибор” (Рисунок 15) приложение должно предоставить пользователю возможность добавить новый прибор. На экране должны быть организованы следующие функциональные требования:

Добавить прибор

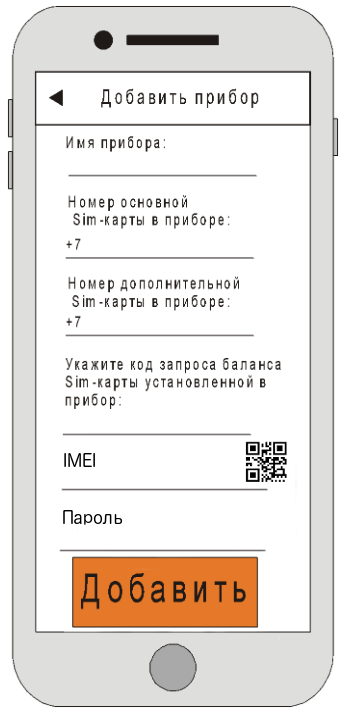
Модель запроса: **CreateDevice**. Поля:

Рисунок - экране “Добавить прибор”

* number - IMEI номер прибора (15 символов)
* name - название прибора
* mainSim - номер основной Sim-карты в приборе
* additionalSim - номер дополнительной Sim-карты в приборе
* balanceCode - код запроса баланса Sim-карты, установленной в прибор
* type - тип прибора, поле типа **DeviceType**

Модель ответа: **CreateDeviceResult**. Поля:

* isCreated - булевый флаг, прибор создан/не создан

**DeviceType.** Перечисление типа:

1 – ИПРО-7

1. Пользователь должен иметь возможность вводить информацию для идентификации прибора: имя прибора, номера основной и дополнительной Sim карты в приборе, код запроса баланса, а также, в случае если приложение находится в режиме онлайн (подробнее об этом в [Принципы работы приложения “ИПРО-7”](#_1y810tw)) - индивидуальный номер прибора (IMEI) и пароль для доступа к прибору.
2. Приложение должно обеспечивать валидацию введенной пользователем информации после нажатия пользователем на кнопку “Добавить”. А также сообщать пользователю о найденных проблемах. Валидация включает в себя:

* проверка ввода пользователем номера основной sim-карты в приборе;
* проверка ввода пользователем индивидуального номера прибора и пароля (в случае если приложение находится в режиме онлайн);
* проверка введенных номеров телефонов – состоят только из цифр, количество цифр должно быть больше или равно 6.

1. Приложение должно обеспечивать сохранения данных, введенных пользователем в локальную БД в таблицу ([Таблица 2. Devices](#_1au1eum)).
2. Приложение должно обеспечивать отправку данных, введенных пользователем на сервер при помощи запроса ([Добавить прибор](#_2gb3jie)) или смс-сообщения ([Запись телефонного номера в телефонную книгу прибора](#_z337ya)).
3. Приложение должно обеспечивать переход приложения на экран [“Отображение и выбор используемого прибора”](#_2lwamvv).

**Экран “Редактировать прибор”**

На экране “Редактировать прибор” (Рисунок 16) приложение должно предоставить пользователю возможность редактировать данные выбранного прибора. На экране должны быть организованы следующие функциональные требования:

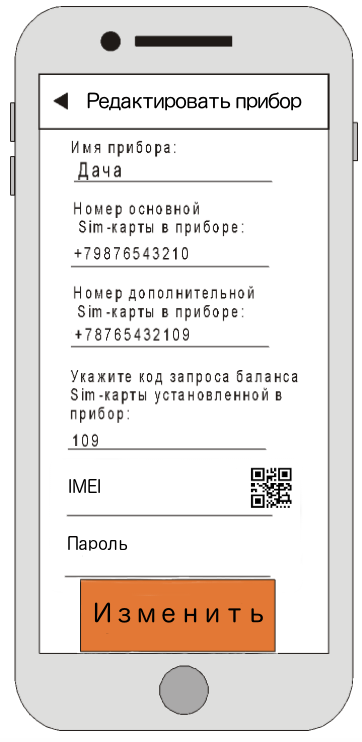
 Изменить прибор

Рисунок - экран “Редактировать прибор”

Модель запроса: **EditDevice**. Поля:

* id - идентификатор прибора
* number - IMEI номер прибора (15 символов)
* name - название прибора
* mainSim - номер основной Sim-карты в приборе
* additionalSim - номер дополнительной Sim-карты в приборе
* balanceCode - код запроса баланса Sim-карты, установленной в прибор
* isMain - булевый флаг, прибор является/не является главным

Модель ответа: **EditDeviceResult**. Поля:

* isSaved - 0 - изменения не сохранены, 1 - изменения сохранены

1. Пользователь должен иметь возможность редактировать информацию о приборе: имя прибора, номера основной и дополнительной Sim карты в приборе, код запроса баланса, а также, в случае если приложение находится в режиме онлайн (подробнее об этом в [Принципы работы приложения “ИПРО-7”](#_1y810tw)) - индивидуальный номер прибора (IMEI) и пароль для доступа к прибору.
2. Приложение должно обеспечивать валидацию введенной пользователем информации после нажатия пользователем на кнопку “Изменить”. А также сообщать пользователю о найденных проблемах. Валидация включает в себя:

* проверка ввода пользователем номера основной sim-карты в приборе;
* проверка ввода пользователем индивидуального номера прибора и пароля (в случае если приложение находится в режиме онлайн);
* проверка введенных номеров телефонов - состоят только из цифр, количество цифр должно быть больше или равно 6.

1. Приложение должно обеспечивать сохранение данных, введенных пользователем в локальную БД в таблицу ([Таблица 2. Devices](#_1au1eum)).
2. Приложение должно обеспечивать отправку данных, введенных пользователем на сервер при помощи запроса ([Изменить прибор](#_1ulbmlt)).
3. Приложение должно обеспечивать переход приложения на экран [“Отображение и выбор используемого прибора”](#_2lwamvv).

**Экран “Температура”**

На экране “Температура” (Рисунок 17) приложение должно предоставить пользователю информацию, передаваемую с термодатчиков прибора. На экране должны быть организованы следующие функциональные требования:

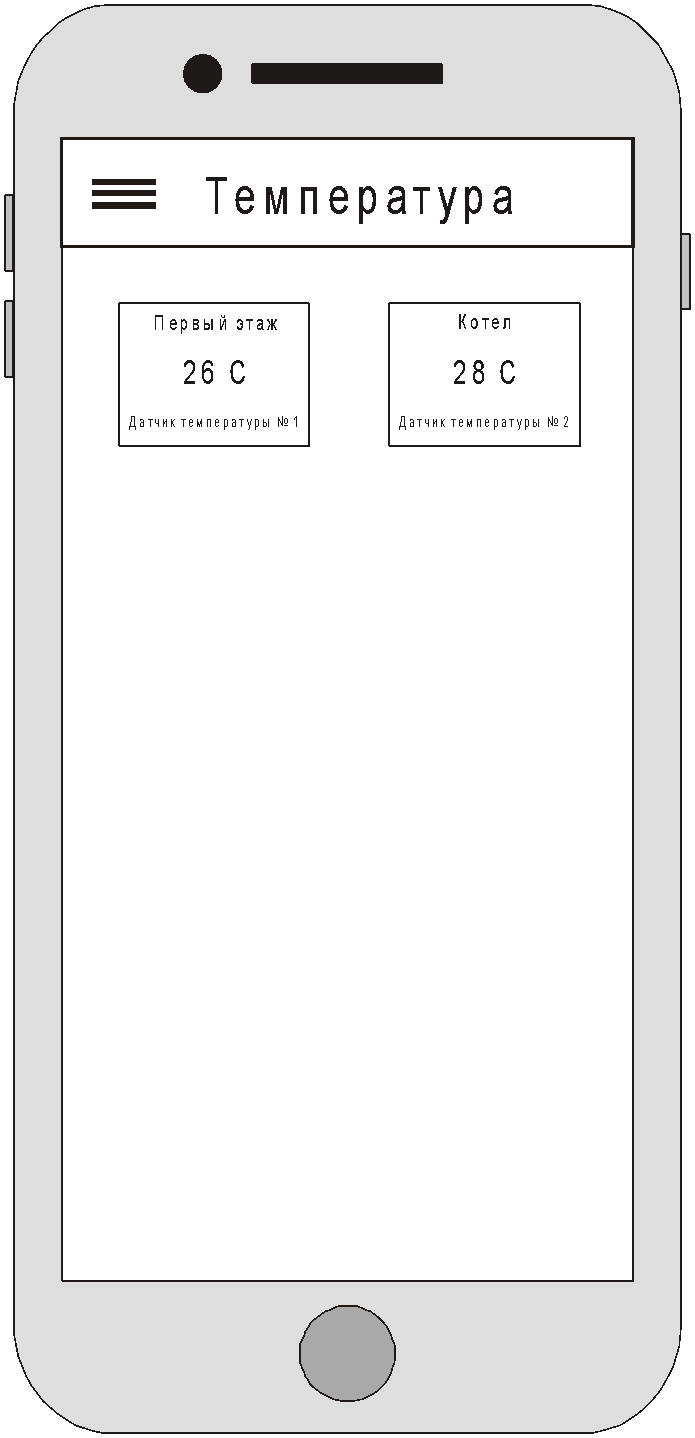
1. Приложение должно обеспечить отображение плиток с информацией о всех термодатчиках, подключенных к прибору. В каждой плитке должна отображаться следующая информация: название датчика, температура, присылаемая с этого датчика, название размещения на приборе (например: температурный датчик 1 и т.д.) В случае, если данных о температуре нет - отображается надпись: “Нет информации”.

Рисунок - экран “Температура”

1. Приложение должно обеспечить переход приложения на экран [“Датчик температуры”](#_37m2jsg) при выборе пользователем соответствующей плитки.

**Экран “Зоны контроля”**

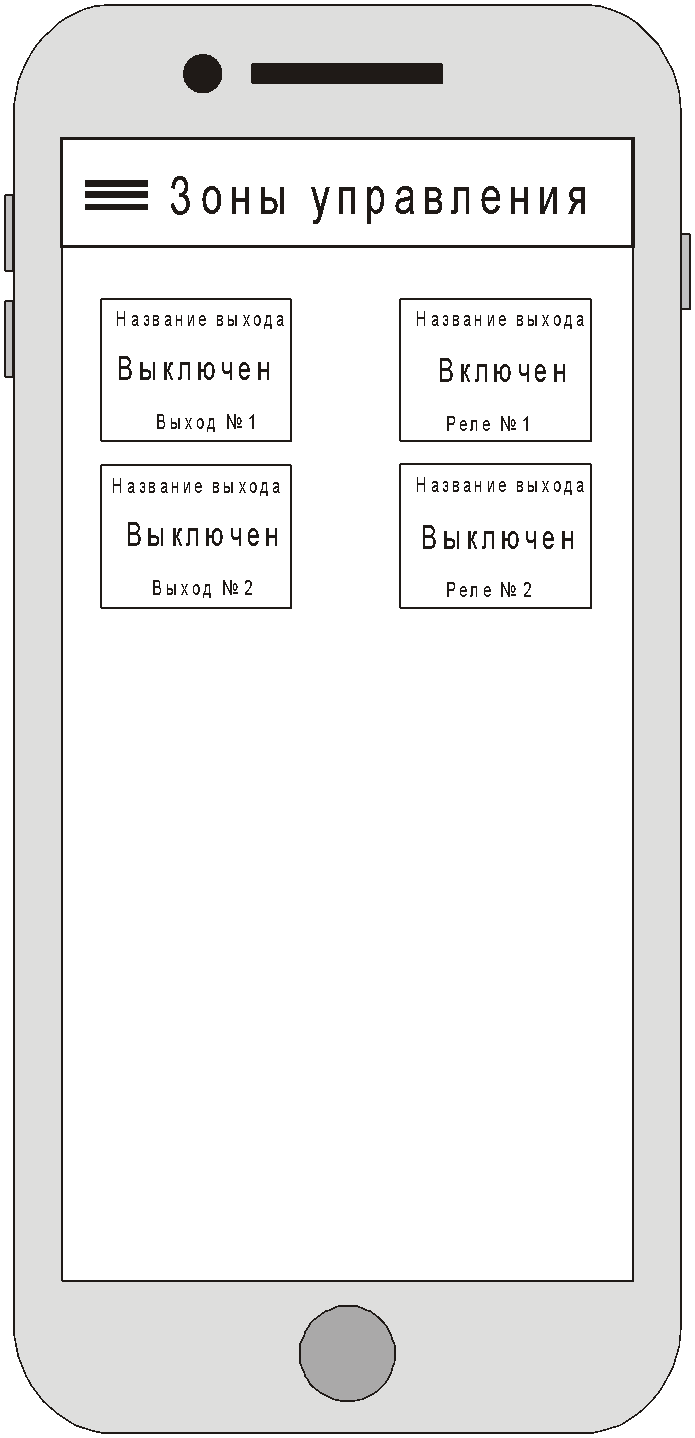
На экране “Зоны контроля” (Рисунок 18) приложение должно предоставить пользователю информацию, передаваемую со всех входов прибора. На экране должны быть организованы следующие функциональные требования:

1. Приложение должно обеспечить отображение плиток с информацией о всех датчиках, подключенных к входам прибора. В каждой плитке должна отображаться следующая информация: название входа, состояние датчика (норма, тревога, отключен), название размещения на приборе (например: беспроводной вход 1, проводной вход 2 и т.д.) В случае, если данных состоянии входа нет - отображается надпись “Нет информации”.

Рисунок - экран “Зоны контроля”

1. Приложение должно обеспечить переход приложения на экран [“Проводной вход”](#_19c6y18) или [“Беспроводной вход”](#_28h4qwu) (в зависимости от выбранной плитки), в котором представлена информация о выбранном входе.

**Экран “Зоны управления”**

На экране “Зоны управления” (Рисунок 19) приложение должно предоставить пользователю информацию, передаваемую со всех выходах прибора. На экране должны быть организованы следующие функциональные требования:

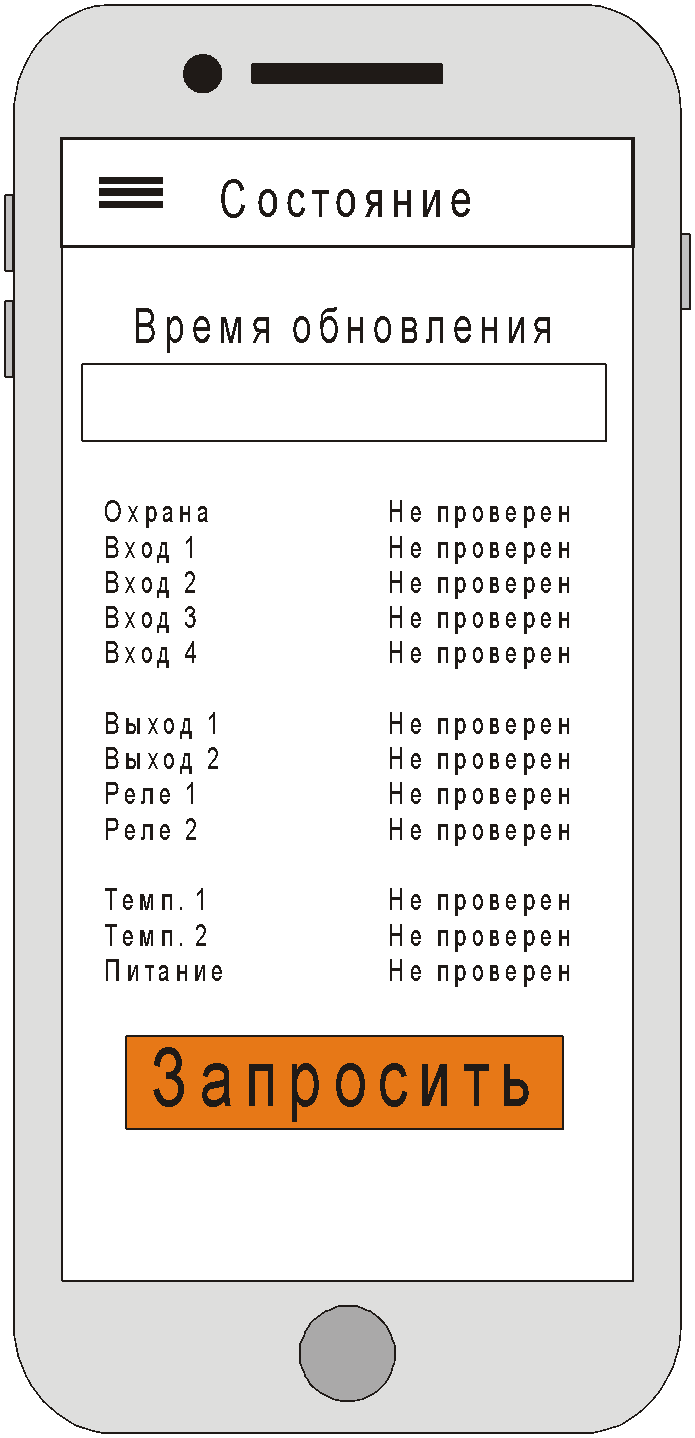
1. Приложение должно обеспечить отображение плиток с информацией о всех датчиках (механизмах), подключенных к выходам прибора. В каждой плитке должна отображаться следующая информация: название выхода, состояние датчика (включен/выключен), название размещения на приборе (например: выход 1, реле 2 и т.д.) В случае, если данных состоянии выхода нет - отображается надпись: “Нет информации”.

Рисунок - экран “Зоны управления”

1. Приложение должно обеспечить переход приложения на экран [“Выход”](#_nmf14n), в котором представлена информация о выбранном входе.

**Экран “Состояние”**

На экране “Состояние” (Рисунок 20) приложение должно предоставить пользователю информацию о состоянии прибора. На экране должны быть организованы следующие функциональные требования:

 Модель ответа: **DeviceState.** Поля:

* id - идентификатор прибора
* isArmed - 0 - прибор снят с охраны, 1 - прибор поставлен на охрану
* inputsState - массив объектов типа **InputState**
* outputsState - массив объектов типа **OutputState**
* sensorState - массив объектов типа **SensorState**
* powerOn - 0 - основное питание отключено, работа от аккумулятора; 1 - нормальное питание от сети
* powerName - название состояния “Основное питание включено”
* powerAkb - название состояния “Питание от АКБ”

Рисунок - экран “Состояние”

* + - * powerOff - название состояния “АКБ разряжен”
      * isPowerMain - 0 - не отображать питание на главном экране, 1 - отображать питание на главном экране

**InputState.** Поля:

* number - номер входа (1, 2, 3 и т.д.)
* name - название входа
* inputType - тип входа, 0 - беспроводной вход, 1 - проводной вход
* state - 0 - вход выключен, 1 - вход включен, 2 - была сработка, вход находится в состоянии Тревога
* normState - название состояния “Норма”
* alarmState - название состояния “Тревога” (текст тревожного сообщения)
* is24 - 0 - “Контроль 24 часа” выключен, 1 - “Контроль 24 часа” включен
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**OutputState.** Поля:

* number - номер выхода (1, 2, 3 и т.д.)
* name - название выхода
* onState - название состояния “Включен”
* offState - название состояния “Выключен”
* isOn - 0 - выход выключен, 1 - выход включен
* time - время на которое был включен выход в секундах
* timeOn - время, когда был включен выход
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**SensorState.** Поля:

* number - номер термодатчика (1, 2, 3 и т.д.)
* name - название термодатчика
* isOn - 0 - термодатчик выключен, 1 - термодатчик включен
* isCorrupted - 0 - термодатчик исправен, 1 - термодатчик неисправен
* setDegree - значение установленной температуры в °С
* actualDegree - значение актуальной температуры в °С
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

1. Приложение должно обеспечить отображение информации о времени последнего запроса о состоянии прибора.
2. Приложение должно обеспечить вывод пользователю всей информации, предоставляемой прибором в виде таблицы - в первом столбце которой отображается название входов, выходов и датчиков прибора, а во втором текущее состояние соответствующего входа, выхода, датчика или прибора в целом.
3. Приложение должно обеспечить обновление информации о состоянии прибора при нажатии пользователя на кнопку “Запросить”. Приложение должно обеспечивать отправку запроса о состоянии прибора введенных на сервер при помощи запроса ([Запросить состояние прибора](#_1xrdshw)) или смс-сообщения [Запрос состояния объекта](#_2et92p0). (Подробнее о запросах к серверу и смс-сообщениях описано в разделе [Принципы работы приложения “ИПРО-7”](#_1y810tw)).

**Экран “История событий”**

На экране “История событий” (Рисунок 21) приложение должно предоставить пользователю историю взаимодействия с прибором (сохраненную в локальной БД). На экране должны быть организованы следующие функциональные требования:

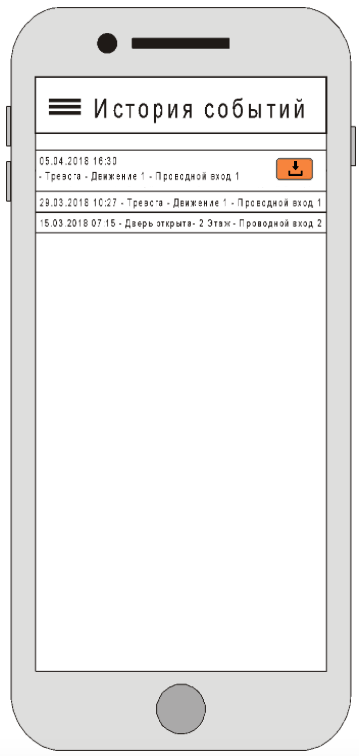
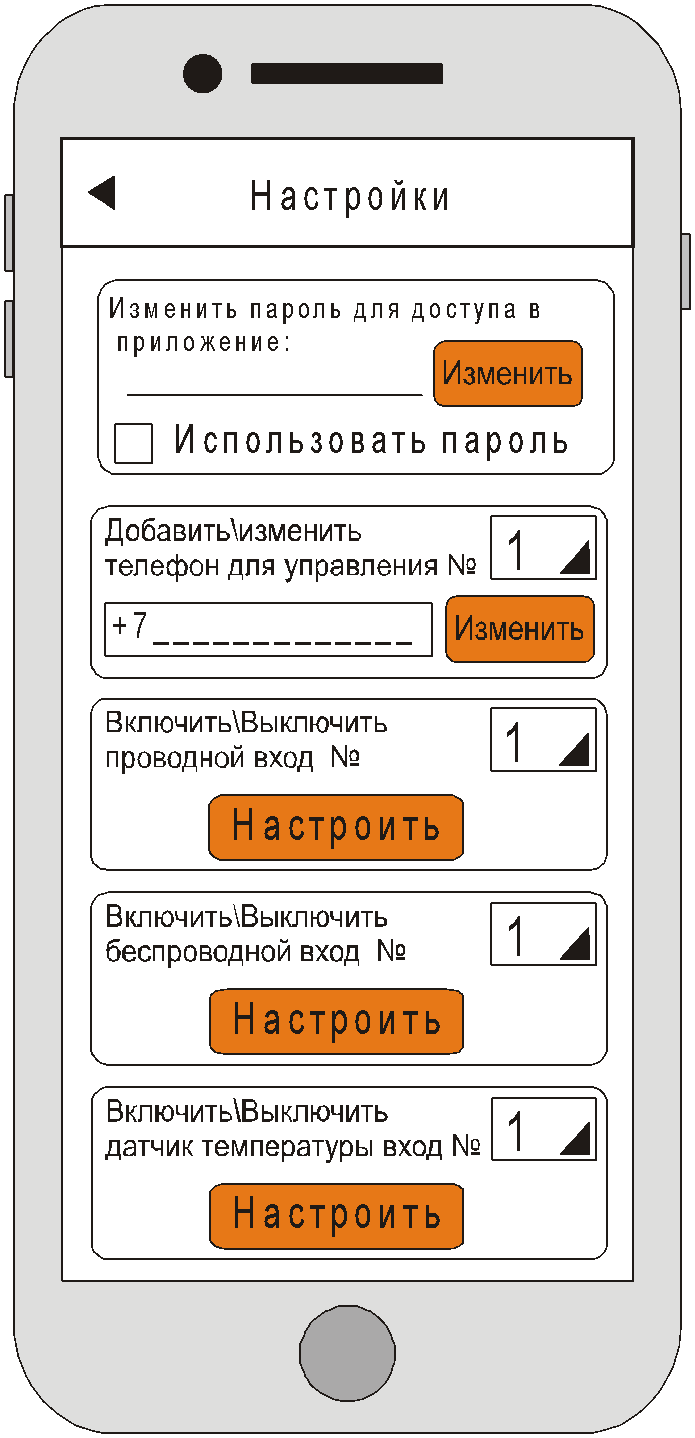
1. Приложение должно обеспечить вывод все, сохраненной в локальной БД, истории взаимодействия с прибором в виде списка. Каждый элемент списка содержит время отправления/получения запроса, название запроса, состояние входа, выхода, датчика, для которого был инициирован запрос.
2. Приложение должно обеспечить возможность очистить историю взаимодействия с прибором. Для этого на экране предусмотрена кнопка “Очистить историю” (расположена справа на тулбаре).

Рисунок - экран “История событий”

**Экран “Настройка”**

На экране “Настройка” (Рисунок 22) приложение должно предоставить пользователю возможность управлять настройками прибора. На экране должны быть организованы следующие функциональные требования:

Добавить/изменить телефонный номер в списке оповещения

Модель запроса:

* id - позиция, на которую будет записываться номер (от 1 до 5)
* phone - номер телефона в международном формате записывается без «+»

Модель ответа: **SavePhone.** Поля:

* isSaved - 0 - номер не записан, 1 - номер записан

1. Пользователь должен иметь возможность включать/выключать проводные и беспроводные входы и датчики температуры. Для этого в приложении должна быть предусмотрена возможность выбрать номер соответствующего входа (термодатчика) и кнопка для включения/выключения выбранного входа (термодатчика).

Рисунок - экран “Настройка”

1. Пользователь должен иметь возможность задать пароль для входа в приложение и определиться, нужно ли спрашивать этот пароль каждый раз при входе в приложение или нет.
2. Приложение должно обеспечить пользователю возможность указать телефон, команды с которого принимает прибор (прибор может запомнить до пяти телефонов). Для этого пользователь может ввести номер телефона в соответствующее поле, выбрать под каким номером записать введенный телефон и отправить изменения на сервер (прибор) путем нажатия на кнопку изменить.
3. Приложение должно предоставить пользователю возможность включить/выключить функцию “контроль 24 часа” для входа.
4. Приложение должно обеспечить переход на экран ["Основные параметры”](#_sqyw64).

**Экран “Настройка Основные параметры”**

На экране “Основные параметры” (Рисунок 23) приложение должно предоставить пользователю возможность управлять общими настройками прибора. На экране должны быть организованы следующие функциональные требования:

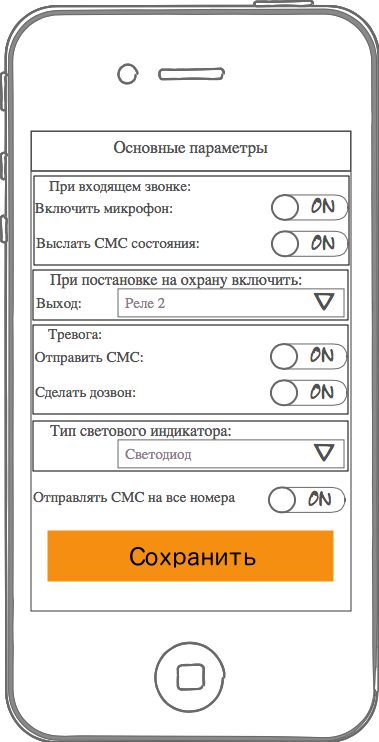


Рисунок - экран “Основные параметры”

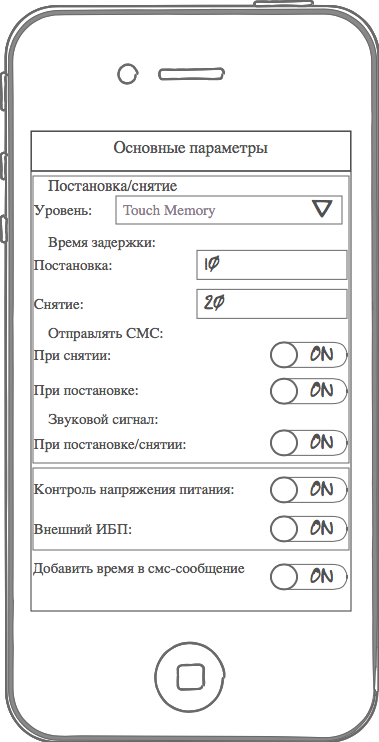
Запросить общие параметры прибора

Параметры запроса:

* id - идентификатор прибора

Модель ответа: **DeviceSettings.** Поля:

* stateLevel - уровень постановки (1-высокий уровень; 2-низкий уровень; 3- TouchMemory / импульс)
* armDelay - время задержки на постановку в секундах (от 0 до 250)
* disarmDelay - время задержки на снятие в секундах (от 0 до 250)
* sendSmsArm - отправка SMS при постановке (0-не отправлять, 1-отправлять)
* sendSmsDisarm - отправка SMS при снятии (0-не отправлять, 1-отправлять)
* sound - звуковой сигнал при постановке\снятии (0-выключить, 1-включить)
* powerControl - контроль напряжения питания (0-выключить, 1-включить)
* externalIbp - использовать внешний ИБП (0-выключить, 1-включить)
* addTimeToSms - добавлять Время в SMS сообщении (0-выключить, 1-включить)
* turnOnMicrophone - при входящем звонке включить микрофон (0-выключить, 1-включить)
* sendSmsState - при звонке выслать SMS состояния (0-выключить, 1-включить)
* outputId - номер выхода, который нужно включать при постановке на охрану; 0 - не включать выход
* alarmSendSms - при тревоге делать отправку SMS (0-выключить, 1-включить)
* alarmCall - при тревоге делать дозвон (0-выключить, 1-включить)
* indicator - тип индикатора состояния (1- светодиод, 2 – внешний индикатор)
* sendSmsToAll - отправка SMS сообщения сразу на все номера (0-выключить, 1-включить)

Сохранить общие параметры прибора

Модель запроса: **EditDeviceSettings.** Поля:

* id - идентификатор прибора
* stateLevel - уровень постановки (1-высокий уровень; 2-низкий уровень; 3- TouchMemory / импульс)
* armDelay - время задержки на постановку в секундах (от 0 до 250)
* disarmDelay - время задержки на снятие в секундах (от 0 до 250)
* sendSmsArm - отправка SMS при постановке (0-не отправлять, 1-отправлять)
* sendSmsDisarm - отправка SMS при снятии (0-не отправлять, 1-отправлять)
* sound - звуковой сигнал при постановке\снятии (0-выключить, 1-включить)
* powerControl - контроль напряжения питания (0-выключить, 1-включить)

Рисунок - экран "Основные параметры"

* externalIbp - использовать внешний ИБП (0-выключить, 1-включить)
* addTimeToSms - добавлять Время в SMS сообщении (0-выключить, 1-включить)
* turnOnMicrophone - при входящем звонке включить микрофон (0-выключить, 1-включить)
* sendSmsState - при звонке выслать SMS состояния (0-выключить, 1-включить)
* outputId - номер выхода, который нужно включать при постановке на охрану; 0 - не включать выход
* alarmSendSms - при тревоге делать отправку SMS (0-выключить, 1-включить)
* alarmCall - при тревоге делать дозвон (0-выключить, 1-включить)
* indicator - тип индикатора состояния (1- светодиод, 2 – внешний индикатор)
* sendSmsToAll - отправка SMS сообщения сразу на все номера (0-выключить, 1-включить)

Модель ответа: **EditDeviceSettingsResult.** Поля:

* isSaved - 0 - параметры прибора не сохранены, 1 - параметры прибора сохранены

1. Приложение должно обеспечить возможность пользователю установить уровень - по которому будет происходить постановка/снятие прибора на охрану (возможны следующие варианты: Touch Memory/импульс, низкий уровень, высокий уровень).
2. Пользователь должен иметь возможность установить время задержки для постановки и снятия прибора - для этого используются соответствующие поля ввода.
3. Пользователь должен иметь возможность включить/выключить следующие функции на приборе:

- отправка смс при снятии прибора с охраны,

- отправка смс при постановке прибора на охрану,

- звуковой сигнал при постановке/снятии прибора с охраны,

- контроль напряжения питания,

- подключить внешний ИБП,

- добавление времени в смс-сообщение,

- включить микрофон при входящем звонке,

- выслать смс-сообщение о состоянии прибора при входящем звонке,

- отправить смс-сообщение при тревоге,

- сделать дозвон при тревоге,

- отправлять смс на все номера.

1. Приложение должно обеспечить возможность пользователю установить тип светового индикатора (возможны следующие варианты: Светодиод, Внешний индикатор).
2. Приложение должно обеспечить пользователю возможность включить/выключить отсылку запроса о состоянии прибора через определенный (так же задаваемый пользователем в соответствующем поле ввода) промежуток времени (промежуток времени должен быть от 10 до 300 секунд).
3. Приложение должно обеспечить валидацию введенных пользователем значений (в поля “Время задержки”) и выдавать сообщение о найденных ошибках, а также сохранять в локальной БД и отправлять соответствующий запрос ([Запросить/сохранить общие параметры прибора](#_2uxtw84)) на сервер или смс-сообщение [Задание общих параметров](#_35nkun2) при нажатии пользователем на кнопку “Сохранить”.
4. Описание работы приложения “ИПРО-7”

Принципы работы приложения “ИПРО-7”

Мобильное приложение «ИПРО-7» для мобильных платформ представляет собой панель контроля и управления приборами сигнализации. Связь с приборами осуществляется посредством сервера, к которому направляются запросы о состоянии прибора и его датчиков. Сервер получает данные о состоянии прибора и отправляет результаты обработки клиенту. Каждый раз, когда пользователь совершает действие, требующее участие сервера и прибора, в приложении проверяется наличие сети интернет. В случае если сеть отсутствует - приложение переходит в режим работы offline. Также режим работы offline используется для приборов, которые не зарегистрированы на сервере - то есть те, на которых не настроена связь с сервером и они не авторизованы под текущим пользователем. Такие приборы пользователь регистрирует путем ввода номера основной (дополнительной) sim-карты, установленной на приборе на экране Регистрация.

Далее в ТЗ описывается только работа приложения в режиме Online.

Главный экран

Приложение должно обеспечить отображение всей информации по прибору на главном экране. Для того, чтобы обеспечить структурированное отображение информации, приложение должно содержать 5 вкладок на главном экране.

**Вкладка “Охрана”**

На вкладке “Охрана” (Рисунок 25) должны быть организованы следующие функциональные требования:

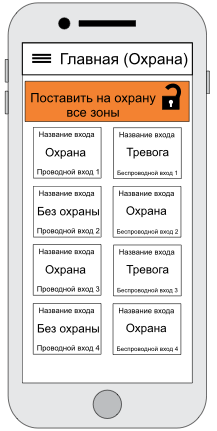
Поставить/снять все зоны с охраны

Рисунок - вкладка “Охрана”

Модель запроса: **Arm**. Поля:

* id - идентификатор прибора
* isArm - 0 - снять прибор с охраны, 1 - поставить прибор на охрану

Модель ответа: **ArmResult**. Поля:

* isArmed - 0 - прибор снят с охраны, 1 - прибор поставлен на охрану

1. Приложение должно обеспечить отображение состояния всех активированных охранных зон прибора. Для этого используются плитки с информацией - каждая плитка содержит название входа, номер входа в устройстве и текущее состояние входа. Для наглядного отображения текущего состояния входа предполагается использовать цветовую индикацию для плиток: “охрана” - желтый, “без охраны” - зеленый, “тревога” - красный.
2. Приложение должно обеспечивать постановку (снятие) всех охранных зон в режим охраны при нажатии пользователем на кнопку “Поставить на охрану” (“Снять с охраны”). Работа приложения в этом случае должна быть обеспечена или запросом к серверу ([Поставить/снять все зоны с охраны](#_18vjpp8)) или отправлением смс-сообщения [Поставить на охрану](https://docs.google.com/document/d/1DYFkZQwDEJQv2bVlqXeqGOwsdsxCmii1Zao8qP7ZGqA/edit?disco=AAAAByv7-QE&ts=5b4387f2#heading=h.u4dy64gvjzqh) ([Снять с охраны](https://docs.google.com/document/d/1DYFkZQwDEJQv2bVlqXeqGOwsdsxCmii1Zao8qP7ZGqA/edit?disco=AAAAByv7-QE&ts=5b4387f2#heading=h.ci8racn0gniq)) непосредственно на прибор (в случае если не удалось связаться с сервером или серверу не удалось связаться с прибором или приложение работает в режиме оффлайн - подробнее это описано в [Принципы работы приложения “ИПРО-7”](#_1y810tw)). Также приложение должно обеспечить информирование пользователя о состоянии прибора (снят с охраны или нет) путем изменения состояния соответствующей кнопки.
3. Приложение должно обеспечить постановку (снятие) в режим охраны отдельного входа путем нажатия пользователя на соответствующую плитку.

**Вкладка “Выходы”**

На вкладке “Выходы” (Рисунок 26) должны быть организованы следующие функциональные требования:

1. Приложение должно обеспечить отображение состояния всех исполнительных выходов прибора. Для этого используются плитки с информацией - каждая плитка содержит название выхода, номер выхода в устройстве и текущее состояние выхода. Для наглядного отображения текущего состояния выхода предполагается использовать цветовую индикацию для плиток: “Выключен” - зеленый, “Включен” - красный.
2. Приложение должно обеспечить включение (выключение) соответствующего выхода путем нажатия пользователя на соответствующую плитку.

Рисунок - вкладка "Выходы"

**Вкладка “Температура”**

На вкладке “Температура” (Рисунок 27) должны быть организованы следующие функциональные требования:

1. Приложение должно обеспечить отображение информации обо всех активированных зонах контроля и управления температурой. Для каждой зоны отображается пользовательское название термодатчика, текущее (последнее записанное в локальную БД) значение температуры на датчике, номер датчика на приборе, а также поддерживаемый режим работы (режимы работы описаны в экране [“Настройка расписания температуры”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.qwbiq3wgbwhh)).
2. Приложение должно обеспечить цветовую индикацию работы зон контроля и управления температурой: красный - отопление работает, зеленый - исполняющий выход выключен (температура находится в пределах нормы).
3. Приложение должно обеспечить возможность изменять температуру, которую необходимо поддерживать в выбранной зоне - для этого пользователь должен иметь возможность задать требуемую температуру путем нажатия на кнопки “+” или “-” (температура задается в диапазоне от -55 до 125 шагом 0,5). При этом режим работы зона меняется на “Ручной”.
4. Приложение должно обеспечить отображение режима работы зоны с помощью кнопки “Режим работы”. При нажатии на эту кнопку приложение должно переключать работу зоны контроля и управления температурой в режим, установленный в настройках ([“Настройка расписания температуры”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.qwbiq3wgbwhh)).

Рисунок - вкладка "Температура"

**Вкладка “Состояние”**

На вкладке “Состояние” (Рисунок 29) должны быть организованы следующие функциональные требования:

Запросить баланс

Параметры запроса:

* id - идентификатор прибора

Модель ответа: **BalanceResult**. Поля:

* balance - значение баланса Sim-карты установленной в прибор

Запросить состояние прибора

Параметры запроса:

* id - идентификатор прибора

Модель ответа: **DeviceState.** Поля:

* isArmed - 0 - прибор снят с охраны, 1 - прибор поставлен на охрану

Рисунок - вкладка “Состояние”

* inputsState - массив объектов типа **InputState**
* outputsState - массив объектов типа **OutputState**
* sensorState - массив объектов типа **SensorState**
* powerOn - 0 - основное питание отключено, работа от аккумулятора; 1 - нормальное питание от сети
* powerName - название состояния “Основное питание включено”
* powerAkb - название состояния “Питание от АКБ”
* powerOff - название состояния “АКБ разряжен”
* isPowerMain - 0 - не отображать питание на главном экране, 1 - отображать питание на главном экране
* networkType - тип сети, в которой находится прибор
* networkSignalLevel - уровень сигнала сети, в которой находится прибор

**InputState.** Поля:

* number - номер входа (1, 2, 3 и т.д.)
* name - название входа
* inputType - тип входа, 0 - беспроводной вход, 1 - проводной вход
* extendedType - расширенный тип входа: 1 - Нормально-замкнутый (НЗ), 2 - Нормально-разомкнутый (НЗР), 3 - Нормально-замкнутый (НЗ) с оконечным резистором, 4 - Аналоговый вход, 5 - Двухпроводный датчик с питанием по шлейфу (резистор X Ом)
* state - 0 - вход выключен, 1 - вход включен, 2 - была сработка, вход находится в состоянии Тревога
* normState - название состояния “Норма”
* alarmState - название состояния “Тревога” (текст тревожного сообщения)
* is24 - 0 - “Контроль 24 часа” выключен, 1 - “Контроль 24 часа” включен
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**OutputState.** Поля:

* number - номер выхода (1, 2, 3 и т.д.)
* name - название выхода
* onState - название состояния “Включен”
* offState - название состояния “Выключен”
* isOn - 0 - выход выключен, 1 - выход включен
* time - время на которое был включен выход в секундах
* timeOn - время, когда был включен выход
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**SensorState.** Поля:

* number - номер термодатчика (1, 2, 3 и т.д.)
* name - название термодатчика
* isOn - 0 - термодатчик выключен, 1 - термодатчик включен
* isCorrupted - 0 - термодатчик исправен, 1 - термодатчик неисправен
* setDegree - значение установленной температуры в °С
* actualDegree - значение актуальной температуры в °С
* mode - режим работы термодатчика, 1 - управлять температурой по жестко заданным порогам, 2 - управлять температурой при снятии с охраны (комфорт), 3 - управлять температурой по расписанию
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

1. Приложение должно обеспечить проверку баланса основной сим карты прибора при нажатии пользователя на кнопку “Запросить баланс”. Работа приложения в этом случае должна быть обеспечена или запросом к серверу ([Запросить баланс](#_3sv78d1)) или отправлением смс-сообщения [Запрос Баланса](https://docs.google.com/document/d/1DYFkZQwDEJQv2bVlqXeqGOwsdsxCmii1Zao8qP7ZGqA/edit?disco=AAAAByv7-QE&ts=5b4387f2#heading=h.y7gl56yt14yj) непосредственно на прибор (в случае если не удалось связаться с сервером или серверу не удалось связаться с прибором или приложение работает в режиме оффлайн - подробнее это описано в [Принципы работы приложения](#_1y810tw)).
2. Приложение должно обеспечивать отображение состояния питания прибора.
3. Приложение должно отображать тип сети, к которой подключен прибор, а также уровень сигнала сети на приборе.
4. Приложение должно отображать - подключен ли прибор к серверу или нет.

**Главное меню приложения**

Главное меню приложения (Рисунок 30) состоит из следующих элементов:

1. Название активного прибора, при нажатии на этот пункт происходит переход на экран [“Отображение и выбор используемого прибора”](https://docs.google.com/document/d/1DYFkZQwDEJQv2bVlqXeqGOwsdsxCmii1Zao8qP7ZGqA/edit?disco=AAAAByv7-QE&ts=5b4387f2#heading=h.k5zdaxsmu8mz).

Рисунок - Главное меню приложения

1. “Главная” - открывает главное окно приложения ([Основной экран](https://docs.google.com/document/d/1DYFkZQwDEJQv2bVlqXeqGOwsdsxCmii1Zao8qP7ZGqA/edit?disco=AAAAByv7-QE&ts=5b4387f2#heading=h.repkw03os4tx)).
2. “Общие параметры” - переход к экрану [“Общие параметры”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.8t5a1hourfix).
3. “Проводные входы” - открывает экран [“Проводные входы”](#_1x0gk37).
4. “Беспроводные входы” - открывает экран [“Беспроводные входы”](#_1baon6m).
5. “Датчики температуры” - открывает экран [“Датчики температуры”](#_zdd80z).
6. “Работа по расписанию” - открывает экран [“Работа по расписанию”](#_48pi1tg).
7. “Журнал событий” - открывает экран [“История событий”](#_3ygebqi).

9) “Настройки” - - открывает экран [“Общие параметры”](#_1jlao46).

10) “Выход” - происходит выход из приложения.

**Экран “Общие параметры”**

На экране “Общие параметры” (Рисунок 31) приложение должно предоставить пользователю возможность управлять общими настройками прибора. На экране должны быть организованы следующие функциональные требования:

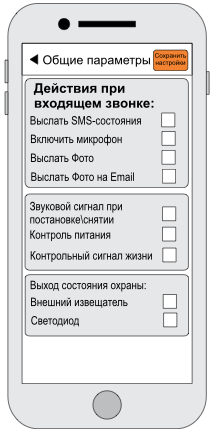
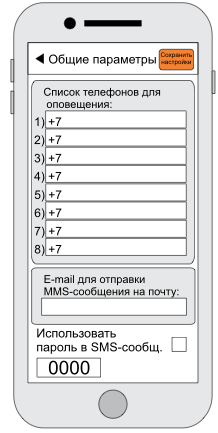


Рисунок - экран “Общие параметры”

Запросить настройки общих параметров прибора

Параметры запроса:

* id - идентификатор прибора

Модель ответа: **Settings**. Поля:

* phones - список телефонов для оповещения, массив объектов типа **PhoneNumber**
* email - E-mail для отправки MMS-сообщения на почту
* isUsePasswordInSms - 0 - не использовать, 1 - использовать пароль в SMS сообщении
* password - пароль для использования в SMS сообщении
* mode - режим постановки/снятия с охраны, 1 - высокий уровень, 2 - низкий уровень, 3 - TouchMemory / импульс
* isSendSmsDuringCall - 0 - выключить, 1 - включить отправку SMS состояния при входящем звонке
* isTurnOnMicrophoneDuringCall - 0 - выключить, 1 - включить микрофон при входящем звонке
* isSendPhotoDuringCall - 0 - выключить, 1 - включить отправку фото при входящем звонке
* isSendPhotoToEmailDuringCall - 0 - выключить, 1 - включить отправку фото на E-mail при входящем звонке
* isPlaySoundSignal - 0 - выключить, 1 - включить звуковой сигнал при постановке\снятии
* isControlPower - 0 - выключить, 1 - включить контроль питания
* isLifeControlSignal - 0 - выключить, 1 - включить контрольный сигнал жизни
* isExternalNotifier - 0 - выключить, 1 - включить внешний извещатель
* isLightDiode - 0 - выключить, 1 - включить светодиод
* sendSmsDuringPhonesArm - список телефонных номеров, на которые нужно отправлять SMS при постановке на охрану, массив объектов типа **PhoneSetting**
* sendMmsDuringPhonesArm - список телефонных номеров, на которые нужно отправлять MMS при постановке на охрану, массив объектов типа **PhoneSetting**
* isSendMmsToEmailArm - 0 - не отправлять, 1 - отправлять фото в виде MMS сообщения на E-mail при постановке на охрану
* isSendMessageToServerArm - 0 - не отправлять, 1 - отправлять тревожное сообщение на сервер при постановке на охрану
* sendSmsIfServerUnavailablePhonesArm - список телефонных номеров, на которые нужно отправлять SMS при постановке на охрану в случае если сервер недоступен, массив объектов типа **PhoneSetting**
* outputIdArm - идентификатор выхода, который нужно включить при постановке на охрану
* isRecordPhotoToSdArm - 0 - не записывать, 1 - записывать фото на SD карту при постановке на охрану
* photosAmountSdArm - количество фото для записи на SD карту при постановке на охрану
* isRecordVideoToSdArm - 0 - не записывать, 1 - записывать видео на SD карту при постановке на охрану
* videoSdArm - время записи видео на SD карту при постановке на охрану в секундах
* sendSmsDuringPhonesDisarm - список телефонных номеров, на которые нужно отправлять SMS при снятии с охраны, массив объектов типа **PhoneSetting**
* sendMmsDuringPhonesDisarm - список телефонных номеров, на которые нужно отправлять MMS при снятии с охраны, массив объектов типа **PhoneSetting**
* isSendMmsToEmailDisarm - 0 - не отправлять, 1 - отправлять фото в виде MMS сообщения на E-mail при снятии с охраны
* isSendMessageToServerDisarm - 0 - не отправлять, 1 - отправлять тревожное сообщение на сервер при снятии с охраны
* sendSmsIfServerUnavailablePhonesDisarm - список телефонных номеров, на которые нужно отправлять SMS при снятии с охраны в случае если сервер недоступен, массив объектов типа **PhoneSetting**
* isRecordPhotoToSdDisarm - 0 - не записывать, 1 - записывать фото на SD карту при снятии с охраны
* photosAmountSdDisarm - количество фото для записи на SD карту при снятии с охраны
* isRecordVideoToSdDisarm - 0 - не записывать, 1 - записывать видео на SD карту при снятии с охраны
* videoSdDisarm - время записи видео на SD карту при снятии с охраны в минутах

**PhoneNumber.** Поля:

* position - позиция, на которую будет записываться номер (1, 2, ...)
* phone - номер телефона в международном формате записывается без «+»

**PhoneSetting.** Поля:

* number - номер телефонного номера в списке оповещения
* isActive - 0 - действие не установлено, 1 - действие установлено для телефона

Сохранить настройки общих параметров прибора

Модель запроса: **Settings**. Поля:

* id - идентификатор прибора
* phones - список телефонов для оповещения, массив объектов типа **PhoneNumber**
* email - E-mail для отправки MMS-сообщения на почту
* isUsePasswordInSms - 0 - не использовать, 1 - использовать пароль в SMS сообщении
* password - пароль для использования в SMS сообщении
* mode - режим постановки/снятия с охраны, 1 - высокий уровень, 2 - низкий уровень, 3 - TouchMemory / импульс
* isSendSmsDuringCall - 0 - выключить, 1 - включить отправку SMS состояния при входящем звонке
* isTurnOnMicrophoneDuringCall - 0 - выключить, 1 - включить микрофон при входящем звонке
* isSendPhotoDuringCall - 0 - выключить, 1 - включить отправку фото при входящем звонке
* isSendPhotoToEmailDuringCall - 0 - выключить, 1 - включить отправку фото на E-mail при входящем звонке
* isPlaySoundSignal - 0 - выключить, 1 - включить звуковой сигнал при постановке\снятии
* isControlPower - 0 - выключить, 1 - включить контроль питания
* isLifeControlSignal - 0 - выключить, 1 - включить контрольный сигнал жизни
* isExternalNotifier - 0 - выключить, 1 - включить внешний извещатель
* isLightDiode - 0 - выключить, 1 - включить светодиод
* sendSmsDuringPhonesArm - список телефонных номеров, на которые нужно отправлять SMS при постановке на охрану, массив объектов типа **PhoneSetting**
* sendMmsDuringPhonesArm - список телефонных номеров, на которые нужно отправлять MMS при постановке на охрану, массив объектов типа **PhoneSetting**
* isSendMmsToEmailArm - 0 - не отправлять, 1 - отправлять фото в виде MMS сообщения на E-mail при постановке на охрану
* isSendMessageToServerArm - 0 - не отправлять, 1 - отправлять тревожное сообщение на сервер при постановке на охрану
* sendSmsIfServerUnavailablePhonesArm - список телефонных номеров, на которые нужно отправлять SMS при постановке на охрану в случае если сервер недоступен, массив объектов типа **PhoneSetting**
* outputIdArm - идентификатор выхода, который нужно включить при постановке на охрану
* isRecordPhotoToSdArm - 0 - не записывать, 1 - записывать фото на SD карту при постановке на охрану
* photosAmountSdArm - количество фото для записи на SD карту при постановке на охрану
* isRecordVideoToSdArm - 0 - не записывать, 1 - записывать видео на SD карту при постановке на охрану
* videoSdArm - время записи видео на SD карту при постановке на охрану в секундах
* sendSmsDuringPhonesDisarm - список телефонных номеров, на которые нужно отправлять SMS при снятии с охраны, массив объектов типа **PhoneSetting**
* sendMmsDuringPhonesDisarm - список телефонных номеров, на которые нужно отправлять MMS при снятии с охраны, массив объектов типа **PhoneSetting**
* isSendMmsToEmailDisarm - 0 - не отправлять, 1 - отправлять фото в виде MMS сообщения на E-mail при снятии с охраны
* isSendMessageToServerDisarm - 0 - не отправлять, 1 - отправлять тревожное сообщение на сервер при снятии с охраны
* sendSmsIfServerUnavailablePhonesDisarm - список телефонных номеров, на которые нужно отправлять SMS при снятии с охраны в случае если сервер недоступен, массив объектов типа **PhoneSetting**
* isRecordPhotoToSdDisarm - 0 - не записывать, 1 - записывать фото на SD карту при снятии с охраны
* photosAmountSdDisarm - количество фото для записи на SD карту при снятии с охраны
* isRecordVideoToSdDisarm - 0 - не записывать, 1 - записывать видео на SD карту при снятии с охраны
* videoSdDisarm - время записи видео на SD карту при снятии с охраны в секундах
* powerName - название состояния “Основное питание включено”
* powerAkb - название состояния “Питание от АКБ”
* powerOff - название состояния “АКБ разряжен”
* isPowerMain - 0 - не отображать питание на главном экране, 1 - отображать питание на главном экране

**PhoneNumber.** Поля:

* position - позиция, на которую будет записываться номер (1, 2, ...)
* phone - номер телефона в международном формате записывается без «+»

**PhoneSetting.** Поля:

* number - номер телефонного номера в списке оповещения
* isActive - 0 - действие не установлено, 1 - действие установлено для телефона

1. Приложение должно обеспечить пользователю возможность заполнить список телефонов, с которыми прибор будет контактировать - для этого предусмотрены 8 полей ввода. Пользователь также должен иметь возможность редактировать уже введенные телефонные номера.
2. Приложение должно обеспечить пользователю возможность задать или отредактировать E-mail адрес для отправки ММS-сообщения.
3. Пользователь должен иметь возможность задать пароль, который будет использоваться при отправке смс-сообщений прибору, а также возможность включить/выключить использование пароля в смс-сообщениях.
4. Приложение должно обеспечить возможность пользователю установить уровень - по которому будет происходить постановка/снятие прибора на охрану (возможны следующие варианты: Touch Memory/импульс, низкий уровень, высокий уровень).
5. Пользователь должен иметь возможность определить действия прибора при входящем звонке, для этого в приложении должны быть предусмотрены переключатели для включения/выключения следующих опций:

* отсылка смс-сообщения о состоянии прибора;
* включение микрофона;
* выслать фото;
* выслать фото на email-адрес.

1. Приложение должно обеспечить пользователю возможность включить/выключить следующие опции:

* звуковой сигнал при постановке/снятии прибора на охрану;
* контроль питания;
* контрольный сигнал жизни;
* При выходе из состояния охраны включить внешний извещатель;
* При выходе из состояния охраны включить светодиод.

1. Приложение должно обеспечить переход на экран [“Действия при постановке на охрану”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.b6ck9tkz17hz) при нажатии на кнопку “Действия при постановке на охрану”.
2. Приложение должно обеспечить переход на экран [“Действия при снятии с охраны”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.104k6gkmmr88) при нажатии на кнопку “Действия при снятии с охраны”.
3. Приложение должно обеспечить сохранение настроек, назначенных пользователем и отсылку их на сервер при нажатии пользователем на кнопку “Сохранить настройки”.

**Экран “Действия при постановке на охрану”**

На экране “Действия при постановке на охрану” (Рисунок 32) приложение должно предоставить пользователю возможность управлять настройками прибора, которые отвечают за поведение прибора при постановке его в режим “Охрана”. На экране должны быть организованы следующие функциональные требования:

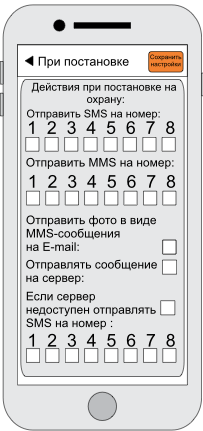
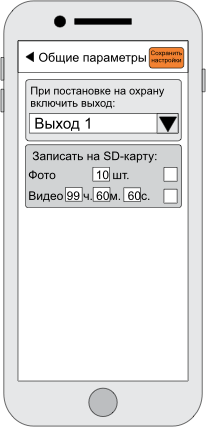
1. Приложение должно обеспечить возможность пользователю настроить на какие номера отправить смс или ммс сообщения при постановке прибора в режим “Охрана”, а также смс-сообщение, в случае если сервер недоступен. При этом для выбора пользователю доступны только заполненные ячейки с номерами телефона.
2. Пользователь должен иметь возможность включить/выключить отправление фотографии в виде ммс-сообщения на email или на сервер.
3. Приложение должно обеспечить пользователю возможность указать какой выход необходимо включать при постановке на охрану.
4. Приложение должно обеспечить пользователю возможность указать какое количество фотографий необходимо записать на SD карту прибора и нужно ли это делать.
5. Приложение должно обеспечить пользователю возможность указать видео какой продолжительностью необходимо записать на SD карту прибора и нужно ли это делать.
6. Приложение должно обеспечить сохранение настроек, назначенных пользователем и отсылку их на сервер при нажатии пользователем на кнопку “Сохранить настройки”.

Рисунок - экран “Действия при постановке на охрану”

**Экран “Действия при снятии с охраны”**

На экране “Действия при постановке на охрану” (Рисунок 33) приложение должно предоставить пользователю возможность управлять настройками прибора, которые отвечают за поведение прибора при снятии его с режима “Охрана”. На экране должны быть организованы следующие функциональные требования:

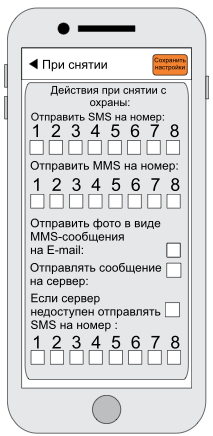
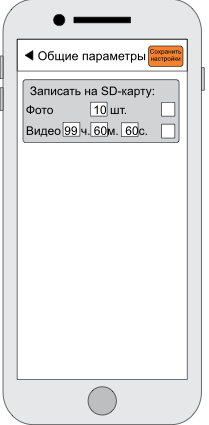
1. Приложение должно обеспечить возможность пользователю настроить на какие номера отправить смс или ммс сообщения при постановке прибора в режим “Охрана”, а также смс-сообщение, в случае если сервер недоступен. При этом для выбора пользователю доступны только заполненные ячейки с номерами телефона.
2. Пользователь должен иметь возможность включить/выключить отправление фотографии в виде ммс-сообщения на email или на сервер.
3. Приложение должно обеспечить пользователю возможность указать какое количество фотографий необходимо записать на SD карту прибора и нужно ли это делать.
4. Приложение должно обеспечить пользователю возможность указать видео какой продолжительностью необходимо записать на SD карту прибора и нужно ли это делать.
5. Приложение должно обеспечить сохранение настроек, назначенных пользователем и отсылку их на сервер при нажатии пользователем на кнопку “Сохранить настройки”.

Рисунок - экран “Действия при постановке на охрану”

**Экран “Проводные входы”**

На экране “Проводные входы” (Рисунок 34) приложение должно предоставить пользователю информацию, передаваемую со всех проводных входов прибора. На экране должны быть организованы следующие функциональные требования:

1. Приложение должно обеспечить отображение состояния всех проводных входов прибора (активированных и неактивированных). Для этого используются плитки с информацией - каждая плитка содержит название входа, номер входа в устройстве и текущее состояние входа.
2. Приложение должно обеспечить переход на экран [“Детали проводного входа”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.bw1tgcf1d4z0) при нажатии пользователя на соответствующую плитку.

Рисунок - экран “Проводные входы”

**Экран “Детали проводного входа”**

На экране “Детали проводного входа” (Рисунок 35) приложение должно предоставить пользователю информацию, о проводном входе прибора. На экране должны быть организованы следующие функциональные требования:

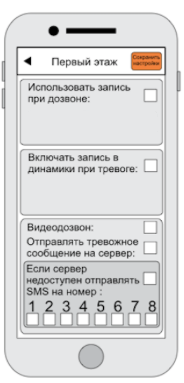
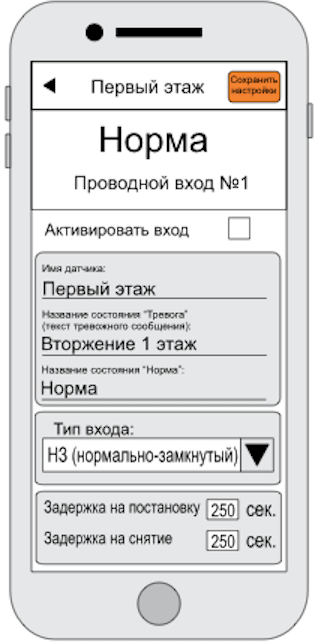




Рисунок - экраны “Детали проводного входа”

Запросить настройки охранного входа

Параметры запроса:

* id - идентификатор прибора
* number - номер входа (1, 2, 3 и т.д.)

Модель ответа: **InputStateResult** Поля:

* name - название входа
* inputType - тип входа, 0 - беспроводной вход, 1 - проводной вход
* extendedType - расширенный тип входа: 1 - Нормально-замкнутый (НЗ), 2 - Нормально-разомкнутый (НЗР), 3 - Нормально-замкнутый (НЗ) с оконечным резистором, 4 - Аналоговый вход, 5 - Двухпроводный датчик с питанием по шлейфу (резистор X Ом)
* state - 0 - вход выключен, 1 - вход включен, 2 - была сработка, вход находится в состоянии Тревога
* normState - название состояния “Норма”
* alarmState - название состояния “Тревога” (текст тревожного сообщения)
* is24 - 0 - “Контроль 24 часа” выключен, 1 - “Контроль 24 часа” включен
* isOn - 0 - не активирован, 1 - вход активирован
* sensorCodes - массив строк - кодов-датчиков (только для беспроводного входа)
* armDelay - задержка на постановку на охрану в секундах
* disarmDelay - задержка на снятие с охраны в секундах
* isAutoArm - 0 - автовзятие выключено, 1 - автовзятие включено
* sendSmsDuringAlarmPhones - список телефонных номеров, на которые нужно отправлять SMS в случае тревоги, массив объектов типа **PhoneSetting**
* sendMmsDuringAlarmPhones - список телефонных номеров, на которые нужно отправлять MMS в случае тревоги, массив объектов типа **PhoneSetting**
* isSendMmsToEmail - 0 - не отправлять, 1 - отправлять фото в виде MMS сообщения на E-mail
* voiceCallPhones - список телефонных номеров, на которые нужно осуществлять голосовой дозвон в случае тревоги, массив объектов типа **PhoneSetting**
* isUseRecordDuringCall - 0 - не использовать, 1 - использовать запись при дозвоне
* isOnRecordDuringAlarm - 0 - не включать, 1 -  включать запись в динамики при тревоге
* isVideoCallOn - 0 - видеодозвон выключен, 1 - видеодозвон включен
* isSendAlarmToServer - 0 - не отправлять, 1 - отправлять тревожное сообщение на сервер
* sendSmsIfServerUnavailablePhones - список телефонных номеров, на которые нужно отправлять SMS в случае если сервер недоступен, массив объектов типа **PhoneSetting**
* isRecordPhotoToSd - 0 - не записывать, 1 - записывать фото на SD карту
* photosAmountSd - количество фото для записи на SD карту
* isRecordVideoToSd - 0 - не записывать, 1 - записывать видео на SD карту
* videoSd - время записи видео на SD карту в минутах
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**PhoneSetting.** Поля:

* number - номер телефонного номера для оповещения (1, 2, 3 и т.д.)
* isActive - 0 - действие не установлено, 1 - действие установлено для телефона

Сохранить настройки охранного входа

Модель запроса: **InputState.** Поля:

* id - идентификатор прибора
* number - номер входа (1, 2, 3 и т.д.)
* name - название входа
* inputType - тип входа, 0 - беспроводной вход, 1 - проводной вход
* extendedType - расширенный тип входа: 1 - Нормально-замкнутый (НЗ), 2 - Нормально-разомкнутый (НЗР), 3 - Нормально-замкнутый (НЗ) с оконечным резистором, 4 - Аналоговый вход, 5 - Двухпроводный датчик с питанием по шлейфу (резистор X Ом)
* state - 0 - вход выключен, 1 - вход включен, 2 - была сработка, вход находится в состоянии Тревога
* normState - название состояния “Норма”
* alarmState - название состояния “Тревога” (текст тревожного сообщения)
* is24 - 0 - “Контроль 24 часа” выключен, 1 - “Контроль 24 часа” включен
* isOn - 0 - не активирован, 1 - вход активирован
* sensorCodes - массив строк - кодов-датчиков (только для беспроводного входа)
* armDelay - задержка на постановку на охрану в секундах
* disarmDelay - задержка на снятие с охраны в секундах
* isAutoArm - 0 - автовзятие выключено, 1 - автовзятие включено
* sendSmsDuringAlarmPhones - список телефонных номеров, на которые нужно отправлять SMS в случае тревоги, массив объектов типа **PhoneSetting**
* sendMmsDuringAlarmPhones - список телефонных номеров, на которые нужно отправлять MMS в случае тревоги, массив объектов типа **PhoneSetting**
* isSendMmsToEmail - булевый флаг, отправлять/не отправлять фото в виде MMS сообщения на E-mail
* voiceCallPhones - список телефонных номеров, на которые нужно осуществлять голосовой дозвон в случае тревоги, массив объектов типа **PhoneSetting**
* isUseRecordDuringCall - 0 - не использовать, 1 - использовать запись при дозвоне
* isOnRecordDuringAlarm - 0 - не включать, 1 -  включать запись в динамики при тревоге
* isVideoCallOn - 0 - видеодозвон выключен, 1 - видеодозвон включен
* isSendAlarmToServer - 0 - не отправлять, 1 - отправлять тревожное сообщение на сервер
* sendSmsIfServerUnavailablePhones - список телефонных номеров, на которые нужно отправлять SMS в случае тревоги, если сервер недоступен, массив объектов типа **PhoneSetting**
* isRecordPhotoToSd - 0 - не записывать, 1 - записывать фото на SD карту
* photosAmountSd - количество фото для записи на SD карту
* isRecordVideoToSd - 0 - не записывать, 1 - записывать видео на SD карту
* videoSd - время записи видео на SD карту в секундах
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**PhoneSetting.** Поля:

* number - номер телефонного номера для оповещения (1, 2, 3 и т.д.)
* isActive - 0 - действие не установлено, 1 - действие установлено для телефона

1. Приложение должно обеспечивать отображение состояния проводного входа прибора (“Тревога”, “Норма”, “Без охраны”, “не активирован”), его пользовательское название и название в приборе.
2. Пользователь должен иметь возможность изменять пользовательское название проводного входа (по умолчанию это название совпадает с названием входа на приборе).
3. Пользователь должен иметь возможность изменять названия для состояния “Тревога” и “Норма” (по умолчанию эти названия: “Тревога вх” + номер входа и “Под охраной”).
4. Пользователь должен иметь возможность активировать/деактивировать вход. Активировать вход - после установки данного флага становятся доступными для изменения все прочие настройки данного входа, и плитка данного входа начинает отображаться на вкладке [“Охрана”](#_1664s55).
5. Пользователь должен иметь возможность определить тип проводного входа, доступны следующие варианты:

* НЗ (нормально-замкнутый)
* НРЗ (нормально-разомкнутый)
* НЗ с оконечным резистором
* Аналоговый вход
* Двухпроводный датчик с питанием по шлейфу (резистор X Ом) (в этом случае приложение автоматически включает переключатель «Контроль 24 часа» в положение “включено” и блокирует возможность установки переключателя «Автовзятие» в положение “включено”).

1. Пользователь должен иметь возможность определить длительность задержки при постановке входа на охрану и при снятии с охраны (временной интервал в пределах от 0 до 250с).
2. Пользователь должен иметь возможность включить/выключить опции “Контроль входа 24 часа” и “Автовзятие”.
3. Приложение должно обеспечить возможность пользователю настроить действия прибора при возникновении “Тревоги”:

* на какие номера прибор должен отправлять смс или ммс сообщения и делать дозвон, нужно ли отправлять фото на почту;
* приложение должно обеспечить включение/отключение проигрывания записи (сделанной пользователем на приборе) при дозвоне;
* приложение должно обеспечить включение/отключение проигрывания записи (сделанной пользователем на приборе) при тревоге;
* приложение должно обеспечить пользователю возможность включить/выключить функцию “Видеодозвон”;
* приложение должно обеспечить пользователю возможность включить/выключить функцию отправки тревожного сообщения через сервер, а также выбрать на какой телефон отправить смс-сообщение, в случае, если в момент возникновения тревоги сервер будет недоступен;
* пользователь должен иметь возможность выбрать в каком виде должны быть зафиксированы события при тревоге (записаны на sd карту прибора) - в виде фото (пользователь может выбрать количество записанных фото в диапазоне от 0 до 10), либо в виде видео (пользователь может определить длительность записанного видео).

1. Приложение должно обеспечить пользователю возможность настроить какие именно выходы должны быть включены при тревоге: для каждого выхода должна быть предусмотрена функция включения/отключения его при возникновении тревоги на этом входе, пользователь также должен указать время, на которое включается выбранный выход или выбрать опцию “включен всегда” (выход включается без учета времени), а также выбрать опцию “включить без задержки” - при тревоге включение выхода происходит без учета времени «задержки снятия».
2. Приложение должно обеспечить сохранение настроек в локальную БД и отправку их на сервер при помощи запроса или смс-сообщения.

**Экран “Беспроводные входы”**

На экране “ Беспроводные выходы” (Рисунок 36) приложение должно предоставить пользователю информацию, передаваемую со всех беспроводных входов прибора. На экране должны быть организованы следующие функциональные требования:

1. Приложение должно обеспечить отображение состояния всех беспроводных входов прибора (активированных и не активированных). Для этого используются плитки с информацией - каждая плитка содержит название входа, номер входа в устройстве и текущее состояние входа.
2. Приложение должно обеспечить переход на экран [“Детали беспроводного входа”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.82yta8z0tyze) при нажатии пользователя на соответствующую плитку.

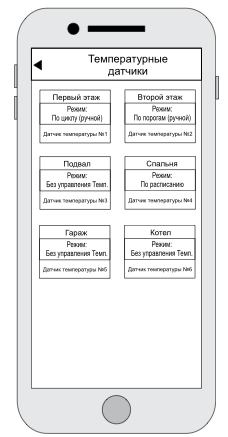
Рисунок - экран “Беспроводные выходы”

**Экран “Детали беспроводного входа”**

Интерфейс экрана “Детали беспроводного входа” полностью аналогичен интерфейсу экрана [“Детали проводного входа”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.yxtxprzzmss).

**Экран “Датчики температуры”**

На экране “Датчики температуры” (Рисунок 37) приложение должно предоставить пользователю информацию, передаваемую со всех температурных датчиков, подключенных к прибору. На экране должны быть организованы следующие функциональные требования:

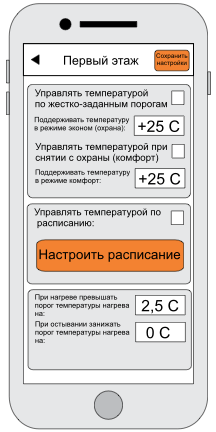
1) Приложение должно обеспечить отображение состояния всех термодатчиков прибора (активированных и не активированных). Для этого используются плитки с информацией - каждая плитка содержит название термодатчика, номер датчика в устройстве в устройстве и текущее состояние датчика (текущий режим работы).

2) Приложение должно обеспечить переход на экран [“Детальный экран датчика температуры”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.jbcgkiy6c2x). при нажатии пользователя на соответствующую плитку.

Рисунок - экран “Датчики температуры”

**Экран “Детальный экран датчика температуры”**

На экране “Детальный экран датчика температуры” (Рисунок 38) приложение должно предоставить пользователю возможность настроить работу выбранного термодатчика. На экране должны быть организованы следующие функциональные требования:



Запросить настройки термодатчика

Параметры запроса:

* id - идентификатор прибора
* number - номер термодатчика (1, 2, 3 и т.д.)

Модель ответа: **TemperatureSensorResult.** Поля:

* name - название термодатчика
* isOn - 0 - термодатчик выключен, 1 - термодатчик включен
* isCorrupted - 0 - термодатчик исправен, 1 - термодатчик неисправен

Рисунок - экран “Детальный экран датчика температуры”

* actualDegree - значение актуальной температуры в °С
* lowDegree - нижний тревожный порог температуры в °С
* highDegree - верхний тревожный порог температуры в °С
* code - код термодатчика
* outputNumber - номер выхода, с помощью которого происходит управление температурой
* mode - режим работы термодатчика, 1 - управлять температурой по жестко заданным порогам, 2 - управлять температурой при снятии с охраны (комфорт), 3 - управлять температурой по расписанию
* modeDegree - температура в °С, которую необходимо поддерживать в выбранном режиме
* upDegree - температура в °С, на которую необходимо превышать порог температуры нагрева при нагреве
* downDegree - температура в °С, на которую необходимо занижать порог температуры нагрева при остывании
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

Сохранить настройки термодатчика

Модель запроса: **EditTemperatureSensor.** Поля:

* id - идентификатор прибора
* number - номер термодатчика (1, 2, 3 и т.д.)
* name - название термодатчика
* isOn - 0 - термодатчик выключен, 1 - термодатчик включен
* lowDegree - нижний тревожный порог температуры в °С
* highDegree - верхний тревожный порог температуры в °С
* code - код термодатчика
* outputNumber - номер выхода, с помощью которого происходит управление температурой
* modeDegree - температура в °С, которую необходимо поддерживать в выбранном режиме
* mode - режим работы термодатчика, 1 - управлять температурой по жестко заданным порогам, 2 - управлять температурой при снятии с охраны (комфорт), 3 - управлять температурой по расписанию
* upDegree - температура в °С, на которую необходимо превышать порог температуры нагрева при нагреве
* downDegree - температура в °С, на которую необходимо занижать порог температуры нагрева при остывании
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

1. Приложение должно обеспечивать отображение текущей температуры (последней записанной в БД) фиксируемой термодатчиком, его пользовательское название и название в приборе, а также код термодатчика.
2. Пользователь должен иметь возможность изменять пользовательское название термодатчика (по умолчанию это название совпадает с названием на приборе).
3. Пользователь должен иметь возможность активировать/деактивировать термодатчик. Активировать термодатчик - после установки данного флага становятся доступными для изменения все прочие настройки данного термодатчика, и плитка данного термодатчика начинает отображаться на вкладке [“Температура”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.m08y8sxilqcf).
4. Пользователь должен иметь возможность задать нижний и верхний тревожные пороги температуры.
5. Пользователь должен иметь возможность задать, с помощью какого выхода будет осуществляться регулировка температуры.
6. Приложение должно предоставить пользователю выбор - каким именно способом будет регулироваться и задаваться температура. Для этого предусмотрено несколько блоков - пользователь может выбрать, какой из блоков должен быть активирован (применен для настроек термодатчика и соответствующего выхода) на данный момент (остальные блоки не активны):

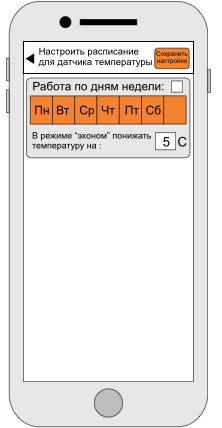
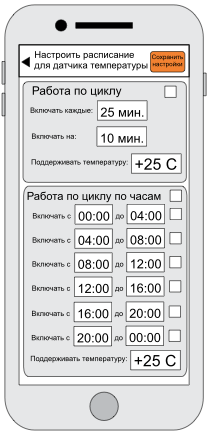
* Управлять температурой по жестко заданным параметрам;
* Управлять температурой по расписанию;

1. В блоке “Управлять температурой по жестко заданным параметрам пользователю должна быть предоставлена возможность ввести температуру, которую прибор будет поддерживать в режиме “Эконом”, а также включить/выключить опцию управления температурой в режиме “комфорт” (после снятия прибора с охраны) и возможность задать температуру для этого режима.
2. Приложение должно иметь возможность перейти к экрану [“Настройка расписания температуры”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.qwbiq3wgbwhh) при нажатии пользователем кнопки “Настроить расписание”.
3. Приложение должно предоставить пользователю возможность ввести на сколько градусов нужно превысить порог температуры нагрева при нагреве, а также ввести насколько градусов можно снижать порог температуры нагрева при остывании.
4. Приложение должно обеспечить валидацию, показ ошибок валидации пользователю, введенной пользователем информации, а также сохранение настроек в локальную БД и отправку их на сервер при помощи запроса или смс-сообщения.

Диапазон выставляемых температур - от -55 до +125.

**Экран “Настройка расписания температуры”**

На экране “Настройка расписания температуры” (Рисунок 39) приложение должно предоставить пользователю возможность настроить работу по расписанию выбранного термодатчика. На экране должны быть организованы следующие функциональные требования:

 Запросить расписание работы термодатчика

Параметры запроса:

* id - идентификатор прибора
* number - номер термодатчика (1, 2, 3 и т.д.)

Модель ответа: Т**emperatureSensorScheduleResult.** Поля:

* mode - режим расписания работы термодатчика, 1 - работа по циклу, 2 - работа по циклу по часам, 3 - работа по дням недели
* degree - поддерживаемая температура в °С

Рисунок - экран “Настройка расписания температуры”

* period - период в мин., через который необходимо включать термодатчик
* interval - интервал в мин., на который необходимо включать термодатчик
* dayPeriods - периоды времени, в которые нужно поддерживать температуру, массив объектов типа **TimePeriod**
* economDegree - значение в °С, на которое необходимо понижать температуру в режиме “эконом”

**TimePeriod.** Поля:

* lowTime - нижняя граница периода
* highTime - верхняя граница периода
* isActive - 0 - не выбран, 1 - выбран данный период

Сохранить расписание работы термодатчика

Модель запроса: **TemperatureSensorSchedule.** Поля:

* id - идентификатор прибора
* number - номер термодатчика (1, 2, 3 и т.д.)
* mode - режим расписания работы термодатчика, 1 - работа по циклу, 2 - работа по циклу по часам, 3 - работа по дням недели
* degree - поддерживаемая температура в °С
* period - период в мин., через который необходимо включать термодатчик
* interval - интервал в мин., на который необходимо включать термодатчик
* dayPeriods - периоды времени, в которые нужно поддерживать температуру, массив объектов типа **TimePeriod**
* economDegree - значение в °С, на которое необходимо понижать температуру в режиме “эконом”

1. Приложение должно предоставить пользователю выбор - каким именно способом будет регулироваться и задаваться температура. Для этого предусмотрено несколько блоков - пользователь может выбрать, какой из блоков должен быть активирован (применен для настроек термодатчика и соответствующего выхода) на данный момент (остальные блоки не активны):

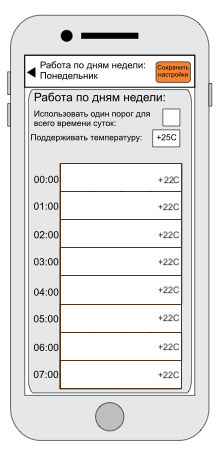
* Работа по циклу;
* Работа по циклу по часам;
* Работа по дням недели.

1. Для блока “Работа по циклу” приложение должно обеспечить пользователю возможность задать промежутки времени (в минутах) (от 5 до 1440), через которые необходимо включать выход, управляющий температурой и на какое время (в минутах) (от 5 до 1440), а также какую температуру необходимо поддерживать. Время работы выхода должно быть меньше чем промежуток срабатывания выхода.
2. Для блока “Работа по циклу по часам” приложение должно обеспечить пользователю возможность активировать в какие (жестко заданные) промежутки времени необходимо включать управляющий выход и какую температуру необходимо поддерживать. Пользователю должны быть доступны следующие временные промежутки:

* с 00.00 до 04.00
* с 04.00 до 08.00
* с 08.00 до 12.00
* с 12.00 до 16.00
* с 16.00 до 20.00
* с 20.00 до 00.00

1. Для блока “Работа по дням недели” приложение должно обеспечить пользователю возможность выбрать каждый из дней недели и перейти на экран [“Дневное расписание температуры”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.syl3ckpny92i). Также должна быть предусмотрена возможность задать на сколько градусов можно понижать температуру в режиме “Эконом”.
2. Приложение должно обеспечить валидацию, показ ошибок валидации пользователю, введенной пользователем информации, а также сохранение настроек в локальную БД и отправку их на сервер при помощи запроса или смс-сообщения.

**Экран “Дневное расписание температуры”**

На экране “Дневное расписание температуры” (Рисунок 40) приложение должно предоставить пользователю возможность настроить работу для дня недели для выбранного термодатчика. На экране должны быть организованы следующие функциональные требования:

Запросить суточное расписание температуры

Параметры запроса:

* id - идентификатор прибора
* number - номер термодатчика (1, 2, 3 и т.д.)
* day - номер дня недели

Модель ответа: **TemperatureSensorDayScheduleResult.** Поля:

* isUseForAllTime - 0 - не использовать, 1 - использовать один порог температуры для всего времени суток

Рисунок - экран “Дневное расписание температуры”

* temperature - массив значений суточной температуры в °С по часам, размер массива 24 элемента (24 часа)

Сохранить суточное расписание температуры

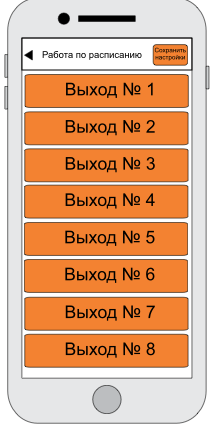
Модель запроса: **TemperatureSensorDaySchedule.** Поля:

* id - идентификатор прибора
* number - номер термодатчика (1, 2, 3 и т.д.)
* day - номер дня недели
* isUseForAllTime - 0 - не использовать, 1 - использовать один порог температуры для всего времени суток
* temperature - массив значений суточной температуры в °С по часам, размер массива 24 элемента (24 часа)

1. Приложение должно обеспечить возможность пользователю задать порог температуры для всего времени суток - для этого, в приложении должно быть предусмотрено поле ввода, а также переключатель. Если пользователь переводит переключатель в положение “включено” - то поля ввода температуры по часам становятся недоступными для ввода. В них автоматически прописывается введенная пользователем в поле “Использовать один порог для всего времени суток”. В случае, если пользователь переводит переключатель в положение “выключено” - поля ввода температуры по часам становятся доступными для ввода. При этом они не очищаются.
2. Пользователь должен иметь возможность ввести для каждого часа температуру, которую необходимо поддерживать. По умолчанию поля не заполняются.
3. Приложение должно обеспечить валидацию, показ ошибок валидации пользователю, введенной пользователем информации, а также сохранение настроек в локальную БД и отправку их на сервер при помощи запроса или смс-сообщения.

**Экран “Работа по расписанию”**

На экране “Работа по расписанию” (Рисунок 41) приложение должно предоставить пользователю возможность выбрать работу какого выхода необходимо настроить. На экране должны быть организованы следующие функциональные требования:



1. Приложение должно обеспечить пользователю отображение всех выходов прибора в виде кнопок и переход на экран [“Расписание для выхода”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.f5wr4wmycxst) при нажатии пользователя на соответствующую кнопку.

Рисунок - экран “Работа по расписанию”

**Экран “Расписание для выхода”**

На экране “Расписание для выхода” (Рисунок 42) приложение должно предоставить пользователю возможность настроить работу по расписанию выбранного выхода прибора. На экране должны быть организованы следующие функциональные требования:



Запросить настройки выхода

Параметры запроса:

* id - идентификатор прибора
* number - номер выхода (1, 2, 3 и т.д.)

Модель ответа: **OutputSettingsResult.** Поля:

* name - название выхода
* onState - название состояния “Включен”
* offState - название состояния “Выключен”
* isOn - 0 - выход выключен, 1 - выход включен

Рисунок - экран “Расписание для выхода”

* mode - режим работы выхода, 1 - работа по циклу, 2 - работа по циклу по часам, 3 - работа по дням недели
* timePeriod - период в мин., через который необходимо включать выход
* timeInterval - интервал в мин., на который необходимо включать выход
* timePeriods - периоды времени, в которые нужно включать выход, массив объектов типа **TimePeriod**
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**TimePeriod.** Поля:

* lowTime - нижняя граница периода
* highTime - верхняя граница периода
* isActive - 0 - не выбран, 1 - выбран данный период

Сохранить настройки выхода

Модель запроса: **OutputSettings.** Поля:

* id - идентификатор прибора
* number - номер выхода (1, 2, 3 и т.д.)
* name - название выхода
* onState - название состояния “Включен”
* offState - название состояния “Выключен”
* mode - режим работы выхода, 1 - работа по циклу, 2 - работа по циклу по часам, 3 - работа по дням недели
* timePeriod - период в мин., через который необходимо включать выход
* timeInterval - интервал в мин., на который необходимо включать выход
* timePeriods - периоды времени, в которые нужно включать выход, массив объектов типа **TimePeriod**
* isMain - 0 - не отображать на главном экране, 1 - отображать на главном экране

**TimePeriod.** Поля:

* lowTime - нижняя граница периода
* highTime - верхняя граница периода
* isActive - 0 - не выбран, 1 - выбран данный период

1. Приложение должно предоставить пользователю выбор - каким именно способом будет регулироваться работы выхода. Для этого предусмотрено несколько блоков - пользователь может выбрать, какой из блоков должен быть активирован (применен для настроек соответствующего выхода) на данный момент (остальные блоки не активны):

* Работа по циклу;
* Работа по циклу по часам;
* Работа по дням недели.

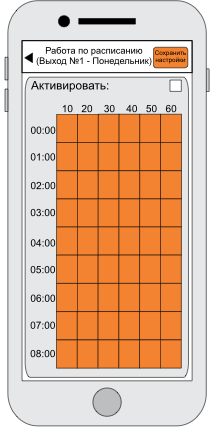
1. Для блока “Работа по циклу” приложение должно обеспечить пользователю возможность задать промежутки времени (в минутах) (от 5 до 1440), через которые необходимо включать выход и на какое время (в минутах) (от 5 до 1440). Время работы выхода должно быть меньше чем промежуток срабатывания выхода.
2. Для блока “Работа по циклу по часам” приложение должно обеспечить пользователю возможность активировать в какие (жестко заданные) промежутки времени необходимо включать выход. Пользователю должны быть доступны следующие временные промежутки:

* с 00.00 до 04.00
* с 04.00 до 08.00
* с 08.00 до 12.00
* с 12.00 до 16.00
* с 16.00 до 20.00
* с 20.00 до 00.00

1. Для блока “Работа по дням недели” приложение должно обеспечить пользователю возможность выбрать каждый из дней недели и перейти на экран [“Дневное расписание”](https://docs.google.com/document/d/1RdAIWxwg03TsPEIfqp4miMLMMSsGzY8QSUiElZ41x1w/edit#heading=h.athbp845618w).
2. Приложение должно обеспечить валидацию, показ ошибок валидации пользователю, введенной пользователем информации, а также сохранение настроек в локальную БД и отправку их на сервер при помощи запроса или смс-сообщения.

**Экран “Дневное расписание”**

На экране “Дневное расписание” (Рисунок 43) приложение должно предоставить пользователю возможность настроить работу для дня недели для выбранного выхода. На экране должны быть организованы следующие функциональные требования:

Запросить суточное расписание работы выхода

Параметры запроса:

* id - идентификатор прибора
* number - номер выхода (1, 2, 3 и т.д.)
* day - день недели

Модель ответа: **OutputScheduleResult.** Поля:

* isActivated - 0 - расписание не активировано, 1 - расписание активировано
* values - матрица (двумерный массив) булевых значений активации выхода в определенный период времени; размер массива 24x6 элементов: 24 часа на 6 периодов времени (0, 10, 20, 30, 40, 50 минут)

Сохранить суточное расписание работы выхода

Рисунок - экран “Дневное расписание”

Модель запроса: **OutputSchedule.** Поля:

* id - идентификатор прибора
* number - номер выхода (1, 2, 3 и т.д.)
* day - день недели
* isActivate - 0 - не активировать расписание, 1 - активировать расписание
* values - матрица (двумерный массив) булевых значений активации выхода в определенный период времени; размер массива 24x6 элементов: 24 часа на 6 периодов времени (0, 10, 20, 30, 40, 50 минут)

1. Приложение должно обеспечить пользователю возможность задать в какое время должен быть активирован выход. Для каждых 10 минут пользователь может определить - должен ли работать выход в эти минуты или нет.
2. Приложение должно обеспечить сохранение настроек в локальную БД и отправку их на сервер при помощи запроса или смс-сообщения.

**3. Предоставляемые Исполнителю Заказчиком документы и материалы.**

Заказчиком Исполнителю предоставляется доступ к разработанному серверному ПО и готовый прибор «ИПРО-7»

**4. Состав и содержание мероприятий.**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование мероприятия** | **Период выполнения (указывается количество календарных дней с даты подписания Договора)** | **Форма результата** |
| **1** | Разработка управляющих программ для управления оборудованием ИПРО-7 | 90 | Рабочее мобильное приложение под IOS и Android в соответствии с ТЗ |

**5. Требования к результатам работ.**

Результатом предоставления услуги является:

Полностью функционирующее мобильное приложение «Умный дом» предназначенное для управления техническими устройствами, их мониторинга и иного взаимодействия в соответствии с техническим заданием.

Исходники мобильных приложений под IOS и Android.

|  |  |  |
| --- | --- | --- |
| Генеральный директор | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (руководитель юридического лица /индивидуальный предприниматель) | (подпись) | (расшифровка подписи) |
|  |  |  |
| М.П. |  | «17» апреля 2020 г. |