## Описание предметной области



Студентам важно запоминать информацию, которую дают на парах. Если человек чувствует себя сонным, то его внимание рассеяно. Когда внимание человека рассеянно он хуже запоминает информацию. Следовательно, если студент не высыпается то он хуже запоминает информацию, поэтому данная программа будет полезна студентам.

## Работники работающие в основном с компьютерами

Работникам работающих с компьютерами важно быть внимательными. Если человек будет чувствовать себя сонным, то он будет не внимателен и может допустить ошибки в заполнении каких-либо данных или при написании программы. Вследствие чего данная программа будет полезна людям работающим на компьютере.

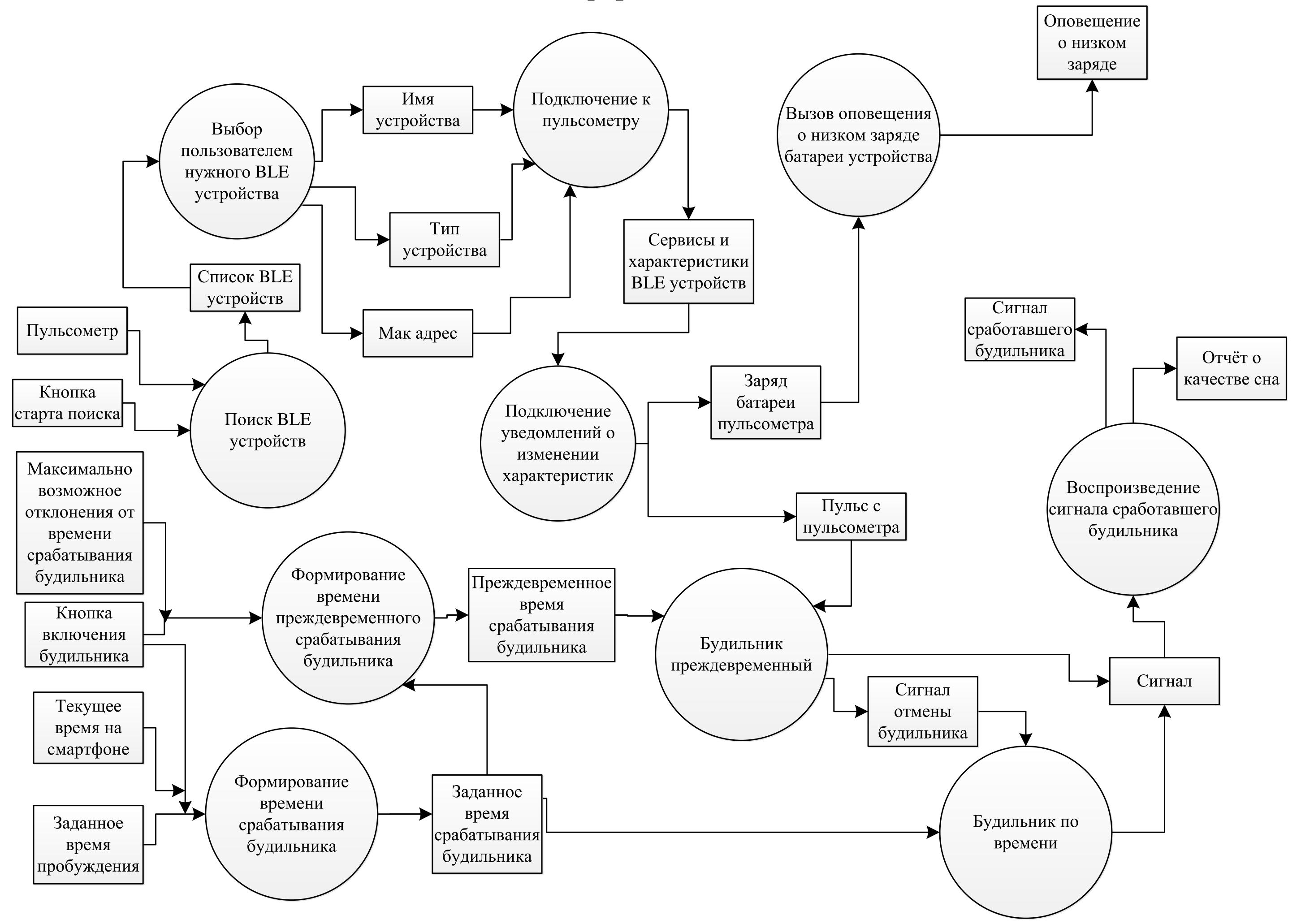


## Работники работ с уклоном в физический труд



Работникам работающих на работах с уклоном на физический труд важно оставаться дольше не уставшим. Если человек чувствует себя сонным, то он быстрее устаёт. Вследствие чего человеку требуется больше времени, для того чтобы отдохнуть после работы. Поэтому данная программа будет полезна людям работающим на работах с уклоном на физический труд

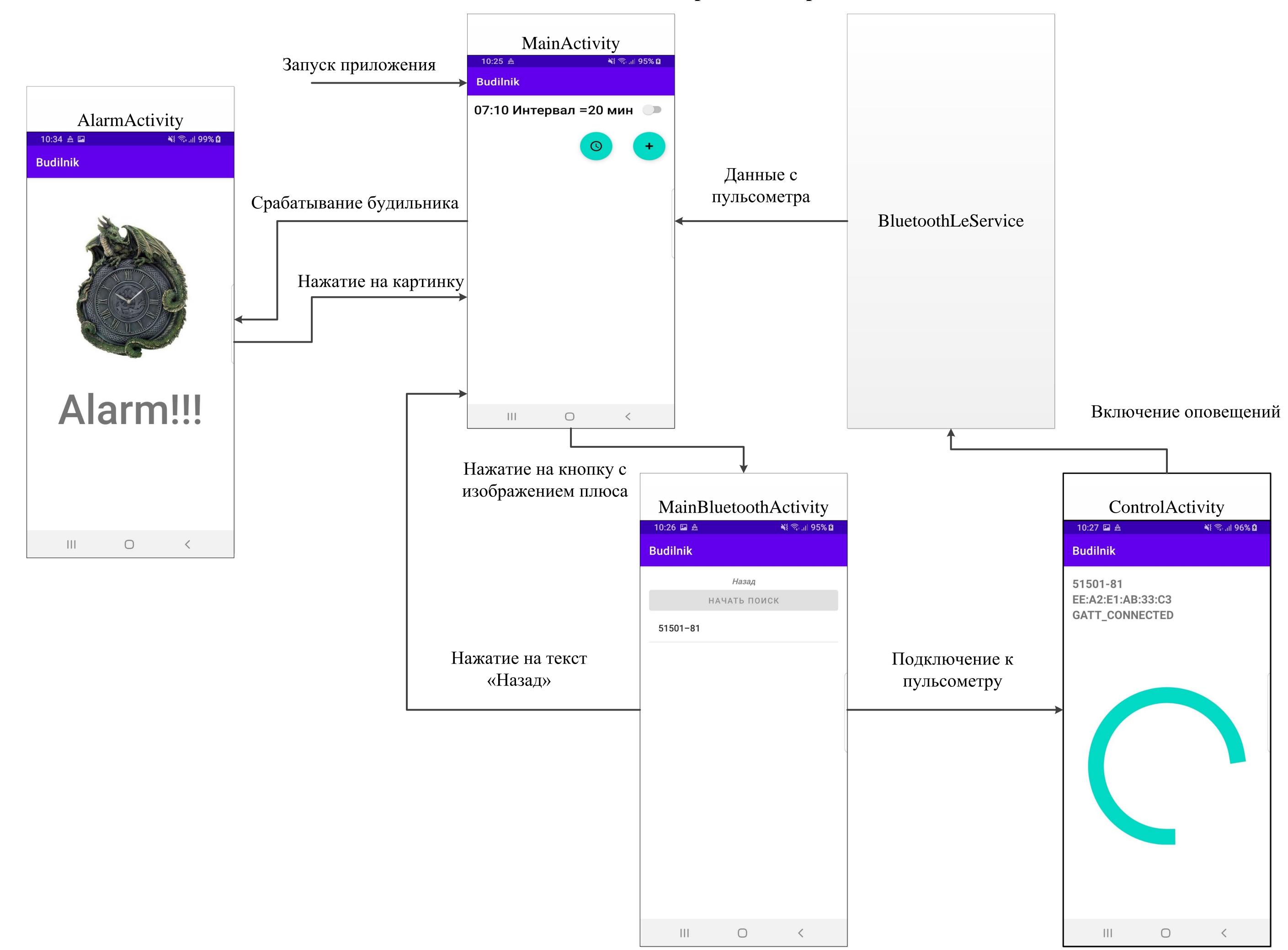
# Схема информационных потоков



# Сравнения аналогов

Название варианта	Изображение устройства	Плюсы	Минусы
Fitbit	OT.OO AM SMT WTFS	Удобное использование	Дороговизна
		Имеются другие функции	Отдельное устройство
			Не определяет фазы сна
Misfit Shine 2		Удобное использование	Дороговизна
		Имеются другие функции	Отдельное устройство
		Определяет фазы сна	
Приложения для отслежывание качества сна	Sleep Score  REM Wake  Sleep Score  REM  Total Sleep Time  Total	Собирает и анализирует	Не выполняет функцию
		информацию о качестве сна	будильника
		Существует на все системы	Не все функции доступны
		смартфонов (Android и IOS)	бесплатно
Приложение Xsmart		Определяет фазы сна	Сложная настройка
		Множество дополнительных	Работает только с браслетом Мі
		функций	Band 2
			Существует только для Android

## Блок схема взаимодействия процессов приложения



## **MainActivity**

## - Отвечает за установку и включение будильника и расчёт минимального времени пробуждения

#### Инициализация переменных:

heartRate;//Переменная хранящая пульс пользователя sdf = new SimpleDateFormat("dd LLLL HH:mm:ss", Locale.getDefault());//Переменная хранящая вормат даты

//Переменные хранящие заданное время будильника public static Integer hour;//Часы

public static Integer min;//Минуты public static Integer day;//День

//Переменные хранящие вычисленное минимальное

время срабатывания будильника public static Integer hour2;//Часы

public static Integer min2;//Минуты public static Integer day2;//День

alarmManager;//AlarmManager нужен для работы будильника

public static final int

REQUEST\_CODE\_LOC=1;Переменная хранящая

switchView;//Переменная хранящая переключатель состояния будильника (вкл.\выкл.) переключательCalendar calendar\_zavtro =

Calendar.getInstance();// Переменная нужная для установки будильника на завтра

calendar\_segodny = Calendar.getInstance();// Переменная нужная для установки будильника на сегодня flagVkl=false;//Переменная хранящая состояние будильника (вкл.\выкл.)

interval;//Переменная хранящая интервал между максимальным временем срабатывания будильника и минимальным

flagVklB=false;//Переменная хранящая состояние будильника (вкл.\выкл.)

#### Методы класса:

onDestroy()//Метод отвечающий за то что произойдёт при переключении с этого класса на другой; onCreate(Bundle savedInstanceState)//Метод вызывающийся при переключении на данный класс fab.onClick(View view)//Метод отвечающий за то что произойдёт при нажатии на кнопку с плюсом Interval2.onClick(View view)//Метод отвечающий за то что произойдёт при нажатии на кнопку с часами switchView.onCheckedChanged(CompoundButton buttonView, boolean isChecked)//Метод отвечающий за то что произойдёт когда переключится переключатель getAlarmInfoPendingIntent()//Установка в нужном виде информации о будильнике

getAlarmActionPendingIntent()//Установка в нужном виде того что произойдёт когда сработает будильник budilka()//Включение будильника otmena()//Отмена включённого будильника zavtro(MaterialTimePicker materialTimePicker,Date currentTime) //установка времени срабатывания

будильника на завтра segodny(MaterialTimePicker materialTimePicker,Date currentTime) // установка времени срабатывания

будильника на сегодня proverka()//Проверка включён ли будильник razrysheniy()//Проверка предоставлены ли приложению

разрешения interval()//Вычисление минимального времени срабатывания будильника

## MainBluetooth

### - Отвечает за поиск BLE устройств Инициализация переменных:

mBluetoothAdapter;//Bluetooth Adapter нужен для работы

с блютузом mBluetoothLeScanner;//Bluetooth Le Scanner нужен для поиска BLE устройств

mScanning;//переменная хранящая состояния поиска (вкл.\выкл.)

RQS\_ENABLE\_BLUETOOTH = 1;//переменная хранящая единицу

btnScan;//переменная хранящая кнопку «Начать поиск» listViewLE;//хранит изображение списка найденных устройств

listBluetoothDevice;//хранит найденные устройства listBluetoothDeviceString;//хранит имена найденных устройств

adapterLeScanResult;//Промежуточная переменная нужная для передачи данных от listBluetoothDeviceString в listViewLE

mHandler;//переменная обработчик процесс поиска устройств

SCAN\_PERIOD = 20000;//сколько максимально будет длиться поиск

### Методы класса:

onCreate(Bundle savedInstanceState)//Метод вызывающийся при переключении на данный класс; OnItemClickListener()//Метод отвечающий за то что произойдёт когда нажмут на имя устройства в списке найденных устройств

getBTDevieType(BluetoothDevice d)Метод определяющий тип устройства getBluetoothAdapterAndLeScanner()//Метод записывающий значения в mBluetoothAdapter и

mBluetoothLeScanner scanLeDevice(final boolean enable)//Метод производящий поиск устройств

onScanResult(int callbackType, ScanResult result)//Метод запускающий запись результатов поиска устройств onBatchScanResults(List<ScanResult> results))//Метод запускающий запись результатов поиска устройств при

onScanFailed(int errorCode)//Метод выводящий код ошибки поиска на экран

addBluetoothDevice(BluetoothDevice device)//Метод записывающий результат поиска BLE устройств oBacking(View view)//Метод отвечающий за то что произойдёт когда нажмут на текст «Назад»

Нажатие на текст «Назад»

пакетном результате

Нажатие на кнопку «На главный экран» или завершение подключения оповещений об изменении значений характеристик устройства

Срабатывание будильника

Нажатие на картинку

### Класс ControlActivity

## - Отвечает за подключение к устройству

#### Инициализация переменных:

Name\_Device;//переменная хранящая статичное имя устройства FILENAME = "LOG.txt";//Названия файла лога входящей информации DIR\_SD = "MyFile";//Название папки в которой хранится лог входящей информации

TAG = "321";//Тэг для нахождения данного класса в логе Android Studio mDeviceName;//переменная хранящая имя устройства

mDeviceAddress;//переменная хранящая адрес устройства

mConnected = false;//Переменная хранящая состояние подключения к

устройству (подключено\не подключено)

mBatteryNotifyCharacteristic;//Переменная хранящая данные для оповещения о изменении характеристики(Заряда батареи устройства)

mPulsNotifyCharacteristic;//Переменная хранящая данные для оповещения о изменении характеристики (Пульс пользователя)

mBluetoothLeService;

textViewState;//Хранит надписи на экране

mGattServicesList;//Хранит перечень сервисов и характеристик устройства

mGattCharacteristics = new

ArrayList<ArrayList<BluetoothGattCharacteristic>>();//Хранит массив характеристик устройства

#### Методы класса:

onServiceConnected(ComponentName componentName, IBinder service)//Метод

обеспечивающий подключение к устройству

onServiceDisconnected(ComponentName componentName)//Метод

вызывающийся при отключении от устройства

onReceive(Context context, Intent intent)//Метод реагирующий на изменение

стадии подключения к устройству

clearUI()//Метод очищающий массив сервисов и характеристик

updateConnectionState(final String st)//Метод выводящий стадию подключения на экран

writeFileSD()//Метод предназначенный для очистки файла LOG

onCreate(@Nullable Bundle savedInstanceState)//Метод вызывающийся при переключении на данный класс

onDestroy()//Метод вызывающийся при переключении с этого класса на другой

oBacking(View view)//Метод вызывающийся при нажатии на кнопку «на главный экран»

BackingGl()//Метод возвращающий на главный экран

Преждевременное срабатывание будильника

## Класс AlarmActivity

### -класс отвечающий за индикацию того что сработал будильник.

### Инициация переменных класса:

//переменная в которой хранится рингтон; //переменная которая управляет вибрацией; vibrator;

// переменная в которой хранятся часы установленного hour; будильника;

// переменная в которой хранятся минуты установленного будильника:

//Переменная хранящая состояние будильника(вкл.\выкл.); flagVkl; switchView; //Переменная отслеживающая на переключатель с главного экрана;

#### Методы класса:

onCreate() //Метод вызывающийся при переключении на данный класс; onDestroy() //Метод вызывающийся при переключении с этого класса на другой;

//Метод выключающий рингтон и вибрацию; cansel()

//Метод отменяющий запланированный будильник если класс otmena() вызвали досрочно;

getAlarmActionPendingIntent()//Метод нужный для определения нужного будильника;

#### Класс BluetoothLeService

## - Отвечает за обработку поступающих с пульсометра данных.

## Инициализация переменных:

batteryPct=0; //Переменная хранящая заряд смартфона

//Тэг для нахождения данного класса в логе Android Studio TAG = "123";DIR\_SD = "MyFile"; //Название папки в которой хранится лог входящей информации

//Названия файла лога входящей информации FILENAME = "LOG.txt"; //Массив для получения входящих данных

//Notification Manager нужен для работы с оповещениями notificationManager;

// Идентификаторы уведомлений

 $NOTIFY_ID = 101;$  $NOTIFYZU_ID = 102;$  $NOTIFYZT_ID = 103;$  $NOTIFYO_ID = 104;$ 

// Название каналов уведомлений

CHANNEL\_ID = "Оповещения о входящих данных";

CHANNELZU\_ID = "Оповещение о малом уровне заряда пульсометра"; CHANNELZT\_ID = "Оповещение о недостаточном уровне заряда телефона"; CHANNELO\_ID = "Оповещение об отключении пульсометра от телефона"; mBluetoothManager; //Bluetooth Manager нужен для работы с блютузом //Bluetooth Adapter нужен для работы с блютузом mBluetoothAdapter;

mBluetoothDeviceAddress; //Хранит адрес блютуз устройства

//Bluetooth Gatt нужен для работы с BLE блютузом mBluetoothGatt;

heartRate; //Хранит полученное значение пульса

mConnectionState = STATE\_DISCONNECTED;//Хранит текущую стадию подключения

//Цифирное обозначение стадий подключения STATE\_DISCONNECTED = 0; //Отключено

STATE\_CONNECTING = 1; //В процессе подключения

STATE CONNECTED = 2; //Подключено

//Переменные нужные для определения стадии подключения к BLE устройству

ACTION\_GATT\_CONNECTED = "android-er.ACTION\_GATT\_CONNECTED";

ACTION\_GATT\_DISCONNECTED = "android-er.ACTION\_GATT\_DISCONNECTED";

ACTION\_GATT\_SERVICES\_DISCOVERED = "android-er.ACTION\_GATT\_SERVICES\_DISCOVERED";

ACTION DATA AVAILABLE = "android-er.ACTION DATA AVAILABLE";

EXTRA DATA = "android-er.EXTRA DATA"; timer=0; //переменная счётчик используется для задержки между оповещениями о входных данных

//UUID Сервисов и характеристик \\Заряд батареи

Battery\_Level\_UUID = UUID.fromString("00002a19-0000-1000-8000-00805f9b34fb");

String\_Name\_ledService = "00001800-0000-1000-8000-00805f9b34fb"; String Name switchChar = "00002a00-0000-1000-8000-00805f9b34fb";

//Пульс UUID Name switchChare = UUID.fromString(String Name switchChar);

String\_Puls\_ledService = "0000180d-0000-1000-8000-00805f9b34fb";

ParcelUuid\_Puls\_ledService = ParcelUuid.fromString(String\_Puls\_ledService);

String\_Puls\_switchChar = "00002a37-0000-1000-8000-00805f9b34fb"; UUID\_Puls\_switchChare = UUID.fromString(String\_Puls\_switchChar);

zaryd=0; //переменная хранящая заряд устройства

//Bluetooth Device нужен для работы с блютуз устройством device; priv=0;

//счётчик использующийся для подтверждения быстрой фазы сна zarydFlag=true; //переменная использующаяся для одиночного срабатывания оповещения о недостатке заряда смартфона zarydFlag2=true; //переменная использующаяся для одиночного срабатывания оповещения о недостатке заряда пульсометра

## Методы класса:

readCharacteristic(BluetoothGattCharacteristic characteristic)//метод для чтения значений из характеристики

writeCharacteristic()//метод для перезаписи названия устройства использующийся для того что бы устройство не «Заснуло» onConnectionStateChange(BluetoothGatt gatt, int status, int newState)//Метод вызывающийся при изменении стадии подключения

onServicesDiscovered(BluetoothGatt gatt, int status)//Метод определяющий что устройство не отключено onCharacteristicRead(BluetoothGatt gatt,BluetoothGattCharacteristic characteristic, int status)//Метод предназначенный для запуска

чтения характеристики onCharacteristicChanged(BluetoothGatt gatt, BluetoothGattCharacteristic characteristic)//Метод определяющий, что характеристика

broadcastUpdate(final String action, final BluetoothGattCharacteristic characteristic)//Метод вызывающийся при изменении

характеристики createChannelIfNeeded(NotificationManager manager)//Метод создающий каналы для оповещений

initialize()//Метод инициализирующий BluetoothManager и BluetoothAdapter;

connect(final String address)//Подключение BLE устройства

close()//Закрытия существующего подключения setCharacteristicNotification(BluetoothGattCharacteristic characteristic, boolean enabled)//Включения оповещения об изменении

getSupportedGattServices() //Находит поддерживаемые устройством сервисы

notifyk()//Оповещения о входящих данных

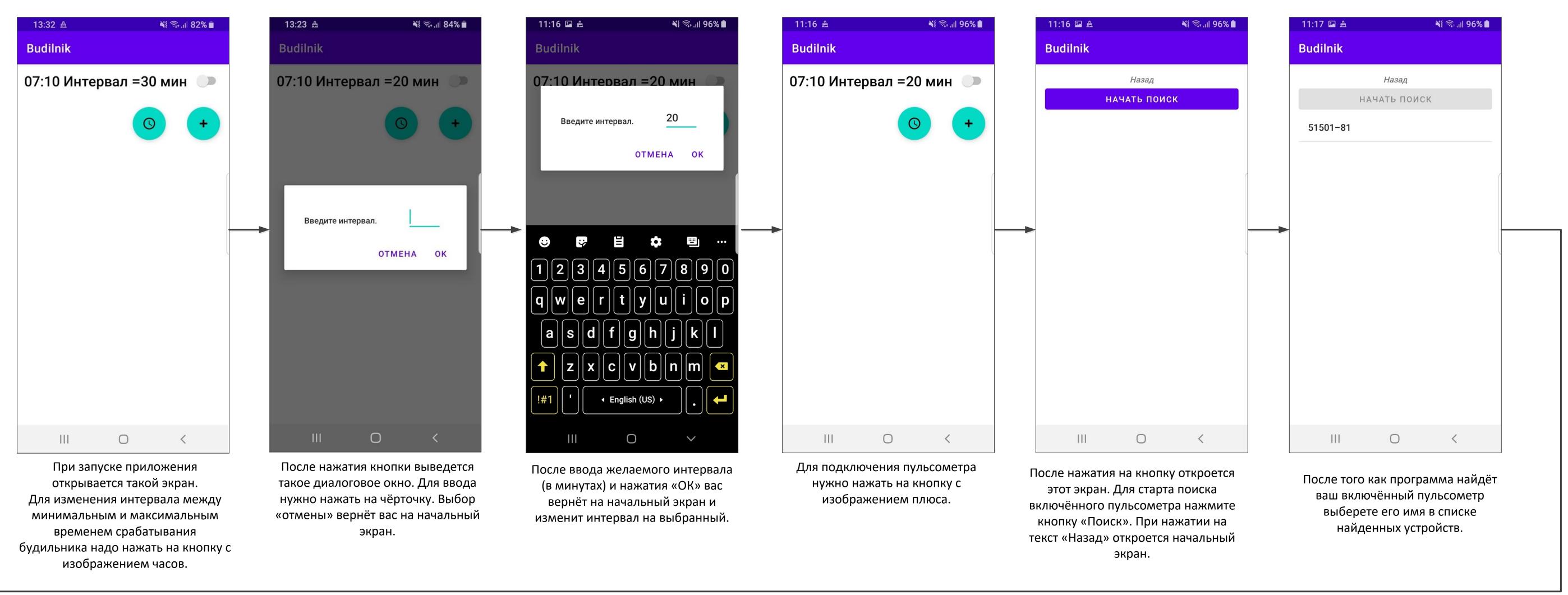
notifykZP()//Оповещение о малом уровне заряда пульсометра

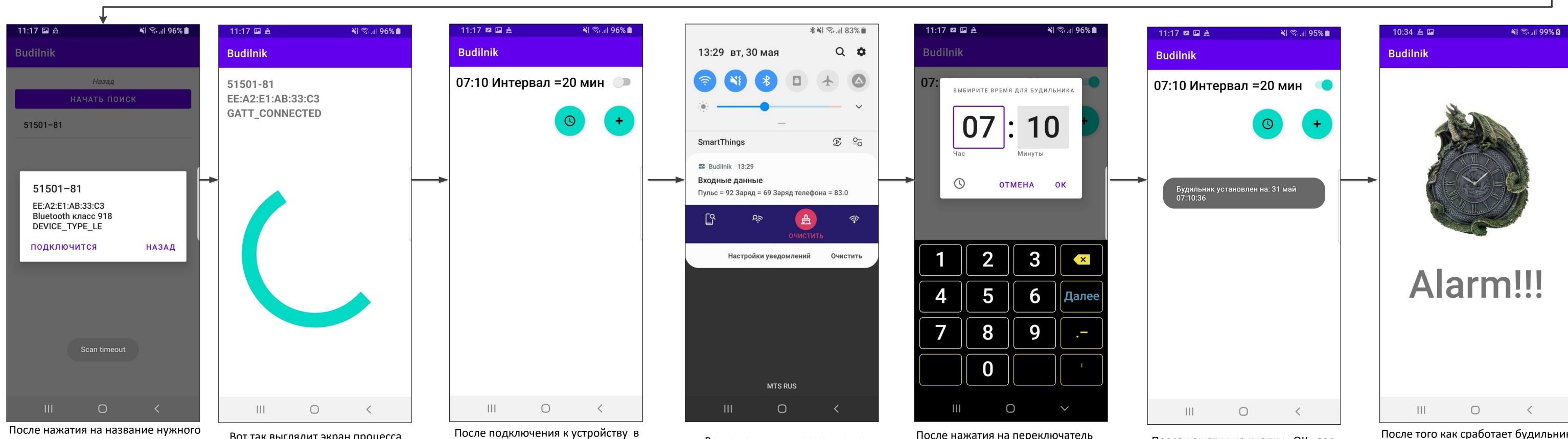
notifykZT()//Оповещение о недостаточном уровне заряда телефона

notifykO()//Оповещение об отключении пульсометра от телефона

budilnik()//Расчёт преждевременного срабатывания будильника

## Интерфейс программы. Контрольный пример.





После нажатия на название нужного устройства появится данное диалоговое окно. При нажатии кнопки «Назад» вас вернёт на экран поиска ВLE Устройств. При нажатии кнопки «Подключится» программа перейдёт на экран процесса подключения к выбранному устройству.

Вот так выглядит экран процесса подключения к устройству. После того как программа подключится к устройству программа вернёт вас на начальный экран, если подключится не получилось, то появится кнопка назад при нажатии на которую вас вернёт к поиску BLE устройств.

уведомлениях телефона появится новое уведомление. Для включения будильника нужно нажать на переключатель находящийся над кнопками.

Вот так выглядит уведомление.

После нажатия на переключатель появится вот такое диалоговое окно для ввода времени срабатывания будильника. Если устанавливаемое время меньше или равно текущему то будильник устанавливается на следующий день.

После нажатия на кнопку «ОК» вас вернёт на начальный экран и напишет на сколько установлен будильник.

После того как сработает будильник откроется этот экран, телефон начнёт вибрировать и если включён начнёт играть музыка рингтона. При нажатии на картинку вас вернёт на начальный экран.

## Статическая модель программы

Нажатие на имя устройства и в открывшемся Нажатие на кнопку с + диалоговом окне на «Подключиться» Подключение оповещений об изменении значений характеристик устройсва Класс BluetoothLeService Класс MainBluetooth Класс ControlActivity - Отвечает за подключение к устройству - Отвечает за обработку поступающих с пульсометра данных. - Отвечает за поиск BLE устройств Инициализация переменных: Инициализация переменных: Инициализация переменных: batteryPct=0; mBatteryNotifyCharacteristic;//Переменная хранящая listBluetoothDevice;//хранит //Переменная хранящая заряд смартфона найденные устройства //Массив для получения входящих данных data; данные для оповещения о изменении Класс MainActivity listBluetoothDeviceString;//хранит характеристики(Заряда батареи устройства) // Название каналов уведомлений - Отвечает за установку и включение CHANNEL\_ID = "Оповещения о входящих данных"; имена найденных устройств mPulsNotifyCharacteristic;//Переменная хранящая будильника и расчёт минимального CHANNELZU\_ID = "Оповещение о малом уровне заряда пульсометра"; adapterLeScanResult;//Промежуточная данные для оповещения о изменении времени пробуждения CHANNELZT\_ID = "Оповещение о недостаточном уровне заряда телефона"; переменная нужная для передачи характеристики (Пульс пользователя) Инициализация переменных: CHANNELO\_ID = "Оповещение об отключении пульсометра от телефона"; данных от listBluetoothDeviceString в mGattServicesList;//Хранит перечень сервисов и heartRate;//Переменная хранящая listViewLE //Хранит полученное значение пульса характеристик устройства heartRate; пульс пользователя mConnectionState = STATE\_DISCONNECTED;//Хранит текущую стадию подключения mHandler;//переменная обработчик sdf = new SimpleDateFormat("dd LLLL //переменная счётчик используется для задержки между Методы класса: процесс поиска устройств timer=0; HH:mm:ss", Locale.getDefault());// SCAN\_PERIOD = 20000;//сколько updateConnectionState(final String st)//Метод оповещениями о входных данных Переменная хранящая вормат даты и максимально будет длиться поиск выводящий стадию подключения на экран //UUID Сервисов и характеристик времени onCreate(@Nullable Bundle savedInstanceState)// Методы класса: //Заряд батареи //Переменные хранящие заданное Battery\_Level\_UUID = UUID.fromString("00002a19-0000-1000-8000-00805f9b34fb"); onCreate(Bundle savedInstanceState)// Метод вызывающийся при переключении на время будильника данный класс //Имя устройства Метод вызывающийся при public static Integer hour;//Часы String\_Name\_ledService ="00001800-0000-1000-8000-00805f9b34fb"; переключении на данный класс; onDestroy()//Метод вызывающийся при public static Integer min;//Минуты String\_Name\_switchChar = "00002a00-0000-1000-8000-00805f9b34fb"; OnItemClickListener()//Метод переключении с этого класса на другой public static Integer day;//День oBacking(View view)//Метод вызывающийся при отвечающий за то что произойдёт //Пульс //Переменные хранящие вычисленное когда нажмут на имя устройства в нажатии на кнопку «на главный экран» UUID\_Name\_switchChare = UUID.fromString(String\_Name\_switchChar); минимальное время срабатывания BackingGl()//Метод возвращающий на главный String\_Puls\_ledService ="0000180d-0000-1000-8000-00805f9b34fb"; списке найденных устройств будильника ParcelUuid\_Puls\_ledService = ParcelUuid.fromString(String\_Puls\_ledService); scanLeDevice(final boolean enable)// экран public static Integer hour2;//Часы Метод производящий поиск String\_Puls\_switchChar = "00002a37-0000-1000-8000-00805f9b34fb"; public static Integer min2;//Минуты UUID\_Puls\_switchChare = UUID.fromString(String\_Puls\_switchChar); устройств public static Integer day2;//День oBacking(View view)//Метод zaryd=0; //переменная хранящая заряд устройства //счётчик использующийся для подтверждения быстрой фазы сна отвечающий за то что произойдёт priv=0; Методы класса: когда нажмут на текст «Назад» Методы класса: onCreate(Bundle savedInstanceState)// broadcastUpdate(final String action, final BluetoothGattCharacteristic characteristic)//Метод Метод вызывающийся при вызывающийся при изменении характеристики переключении на данный класс Преждевременное createChannelIfNeeded(NotificationManager manager)//Метод создающий каналы для fab.onClick(View view)//Метод срабатывание будильника оповещений отвечающий за то что произойдёт при setCharacteristicNotification(BluetoothGattCharacteristic characteristic, boolean enabled)// Нажатие на текст «Назад» нажатии на кнопку с плюсом Включения оповещения об изменении характеристики Interval2.onClick(View view)//Метод notifyk()//Оповещения о входящих данных отвечающий за то что произойдёт при Нажатие на кнопку «На главный экран» Класс AlarmActivity notifykZP()//Оповещение о малом уровне заряда пульсометра или завершение подключения -класс отвечающий за индикацию того что нажатии на кнопку с часами notifykZT()//Оповещение о недостаточном уровне заряда телефона сработал будильник. switchView.onCheckedChanged(Compo оповещений об изменении значений notifykO()//Оповещение об отключении пульсометра от телефона undButton buttonView, boolean Инициация переменных класса: характеристик устройства budilnik()//Расчёт преждевременного срабатывания будильника isChecked)//Метод отвечающий за то // переменная в которой хранятся часы hour; Срабатывание будильника что произойдёт когда переключится установленного будильника; // переменная в которой хранятся переключатель min; Нажатие на картинку budilka()//Включение будильника минуты установленного будильника; otmena()//Отмена включённого Методы класса: onCreate() //Метод вызывающийся при будильника переключении на данный класс; cansel() //Метод выключающий рингтон и вибрацию;

//Метод отменяющий

запланированный будильник если класс вызвали

otmena()

досрочно;