

## Создание приложения будильник на android

*Ульянов Егор Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье рассматривается создания мобильного приложения будильник для android устройства. Разработка происходит в среде разработке android studio. Практическим результатом является рабочее мобильное приложение.

**Ключевые слова:** андроид, будильник, приложение

## Creating an alarm clock app for android

*Ulianov Egor Andreevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

This article discusses how to create a mobile alarm clock application for an android device. Development takes place in the android studio development environment. The practical result is a working mobile application.

**Keywords:** android, alarm clock, app

Будильники существуют еще с 18 века, для людей они оказывают немаленькую помощь. Ведь проспать на работу или на дела никто не хотел. Даже с развитием технологий будильник внедряется во все новые устройства.

Планируется, что это приложение для будильника будет минималистичным и удобным. Можно будет установить будильник для одного события на ближайший день. При срабатывании пользователь будет получать звук будильника, сообщение с уведомлением и сообщение в пользовательском интерфейсе приложения. Это приложение и устройство могут находиться в режиме ожидания или в спящем режиме при срабатывании будильника.

Цель данной статьи создать приложение будильник для android устройств.

В своей работе К.В.Рубинов протестировал большое множество интерфейсов в мобильных приложениях и разработал свой подход к тестированию [1]. А.А. Сковорода и Д.Ю. Гамаюнов рассмотрели множество мобильных приложений с вредоносной базой и предложили методы для автоматической классификации приложений [2]. С.В. Глазков и А.Л. Ронжин

рассмотрели способы формализации и анализа контекста приложений, исполняемых на мобильных гетерогенных устройствах [3]. В.Г.Орлов и С.С. Шаврин рассмотрели мобильные приложения для использования в системах мониторинга и диспетчеризации технологических служб [4].

Будет использоваться «AlarmManagerAPI» для установки и вызова уведомления. Так же будет компонент «TimePicker» и тумблер в пользовательском интерфейсе для установки времени будильника.

Нужно дать разрешение на использование «WAKE\_LOCK», кроме того, что «AndroidManifest.xml» является довольно стандартным и практически больше не изменяется. Далее необходимо включить сервис и ресивер (рис.1).

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

<application android: allowBackup = "true"
android: icon = "@ drawable / ic_launcher"
android: label = "@ string / app_name"
android: theme = "@ style / AppTheme" >
  <activity android: name = ".AlarmActivity"
android: label = "@ string / app_name" > <intent-filter>
    <action android: name = "android.intent.action.MAIN" />
    <category android: name = "android.intent.category.LAUNCHER" />
  </intent-filter> </activity>
  <service android: name = ".AlarmService" android: enabled = "true" />
  <Receiver android: name = ".AlarmReceiver" />
</application>
</manifest>
```

Рисунок 1 - AndroidManifest

Далее переходим в «activity\_main». «Android Activity» разработан, чтобы сделать приложение простым. Есть компонент «TimePicker», за которым следует «ToggleButton». Выберите время для установки будильника и включите переключатель (рис.2).

```
<TimePicker android: layout_width = "wrap_content"
android: layout_height = "wrap_content"
android: id = "@ + id / alarmTimePicker"
android: layout_alignParentTop = "true"
android: layout_centerHorizontal = "true" />

<ToggleButton android: layout_width = "wrap_content"
android: layout_height = "wrap_content"
android: text = "Alarm On / Off"
android: id = "@ + id / alarmToggle"
android: layout_centerHorizontal = "true"
android: layout_below = "@ + id / alarmTimePicker "
android: onClick = " onToggleClicked " />

<TextView android: layout_width = "wrap_content"
android: layout_height = "wrap_content"
android: textAppearance = "? Android: attr / textAppearanceLarge"
android: text = "" android: id = "@ + id / alarmText"
android: layout_alignParentBottom = " true "
android: layout_centerHorizontal = " true "
android: layout_marginTop = " 20dp "
android: layout_below = " @ + id / alarmToggle " />
```

Рисунок 2 - activity\_main

Теперь необходимо связать все написанные кнопки. «AlarmActivity» использует «AlarmManager» для установки сигнала и отправки уведомления при срабатывании сигнала будильника (рис.3).

```
public static AlarmActivity instance() {
    return inst;
}

@Override
public void onStart() {
    super.onStart();
    inst = this;
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_my);
    alarmTimePicker = (TimePicker) findViewById(R.id.alarmTimePicker);
    alarmTextView = (TextView) findViewById(R.id.alarmText);
    ToggleButton alarmToggle = (ToggleButton) findViewById(R.id.alarmToggle);
    alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
}

public void onToggleClicked(View view) {
    if (((ToggleButton) view).isChecked()) {
        Log.d("MyActivity", "Alarm On");
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
        calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());
        Intent myIntent = new Intent(AlarmActivity.this, AlarmReceiver.class);
        pendingIntent = PendingIntent.getBroadcast(AlarmActivity.this, 0, myIntent, 0);
        alarmManager.set(AlarmManager.RTC, calendar.getTimeInMillis(), pendingIntent);
    } else {
        alarmManager.cancel(pendingIntent);
        setAlarmText("");
        Log.d("MyActivity", "Alarm Off");
    }
}
```

Рисунок 3 - AlarmActivity

«AlarmReceiver» - «WakefulBroadcasReceiver» это то, что получает сигнал тревоги в установленное время. Отсюда иницируем различные действия, чтобы уведомить пользователя в нужное время. Предоставим три типа уведомлений: первый показывает сообщение пользователю в пользовательском интерфейсе активности, второй воспроизводит мелодию сигнала будильника и третий отправляет сообщение уведомления Android (рис.4).

```
public class AlarmService extends IntentService {
    private NotificationManager alarmNotificationManager;

    public AlarmService() {
        super("AlarmService");
    }

    @Override
    public void onHandleIntent(Intent intent) {
        sendNotification("Проснись!!!");
    }

    private void sendNotification(String msg) {
        Log.d("AlarmService", "Preparing to send notification...: " + msg);
        alarmNotificationManager = (NotificationManager) this
            .getSystemService(Context.NOTIFICATION_SERVICE);

        PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
            new Intent(this, AlarmActivity.class), 0);

        NotificationCompat.Builder alarmNotificationBuilder = new NotificationCompat.Builder(
            this).setContentTitle("Alarm").setSmallIcon(R.drawable.ic_launcher)
            .setStyle(new NotificationCompat.BigTextStyle().bigText(msg))
            .setContentText(msg);

        alarmNotificationBuilder.setContentIntent(contentIntent);
        alarmNotificationManager.notify(1, alarmNotificationBuilder.build());
        Log.d("AlarmService", "Notification sent.");
    }
}
```

Рисунок 4 - AlarmReceiver

Далее осталось проверить работу будильника (рис.5).

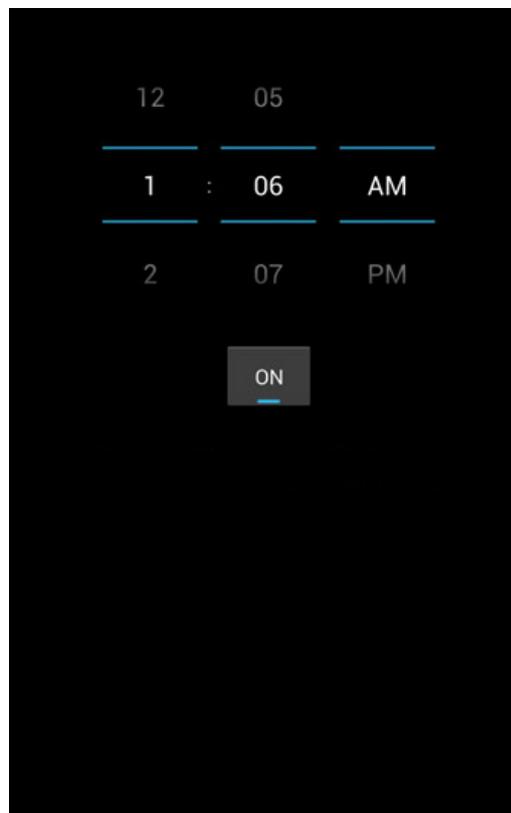


Рисунок 5 – рабочий будильник

В данной статье был описан метод создания мобильного приложения будильник на android-устройства.

### **Библиографический список**

1. Рубинов К.В. Подход к тестированию программных интерфейсов приложений мобильных устройств // Труды Международного симпозиума «Надежность и качество». 2019. №5. С. 16-22.
2. Сковорода А.А., Гамаюнов Д.Ю. Анализ мобильных приложений с использованием моделей привилегий и api-вызовов вредоносных приложений// Современные научные исследования и инновации. 2018. № 3-5 (63). С. 42-44.
3. Глазков С.В., Ронжин А.Л. Методы анализа контекста приложений в мобильных гетерогенных устройствах// Автоматика. Вычислительная техника. 2018. №3. С. 34-39
4. Орлов В.Г., Шаврин С.С. Беспроводные мобильные приложения в системах мониторинга и диспетчеризации технологических служб // Автоматика. Вычислительная техника. 2019. №1. С. 67-71