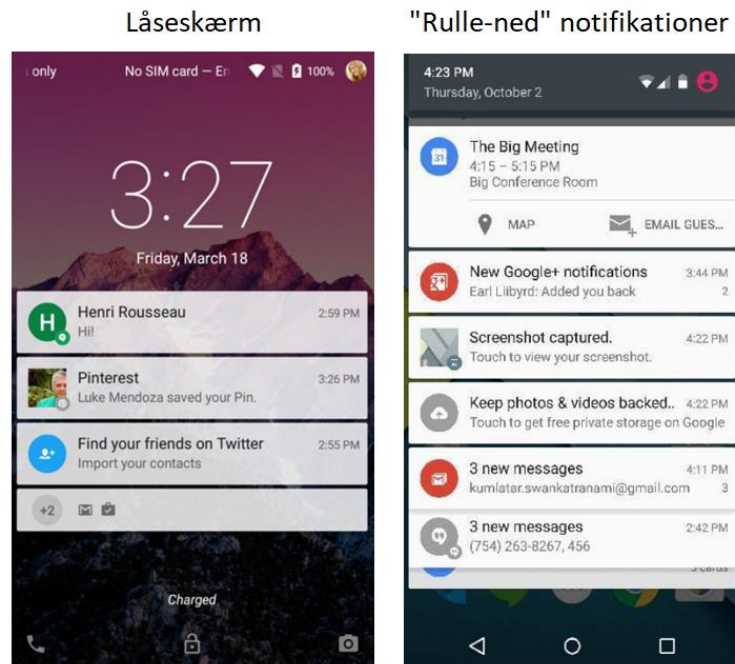


## Artikel 2 (notifikations)

### Introducer teknologien og forklar det som en bruger intet ved om den.

En notifikation er en besked som sendes fra appen til ejeren af smarttelefonen. Beskeden vises ved at brugeren ruller ned fra toppen af skærmen ved brug af fingeren. Herefter kan brugeren trykke på notifikationen for at blive videresendt (typisk til appen eller en hjemmeside). En standard notifikation indeholder minimum et ikon, en titel og noget indhold af tekst. Notifikationer vises også på låseskærmen. Eksempel på notifikationer nedenfor:



### Forklar formålet af teknologien, og hvilke fordele der er

Meningen med en notifikation er så brugeren af appen kan følge med i noget der sker i appen. F.eks. vil Facebook sende en notifikation når nogen skriver en kommentar på din status. Et andet eksempel er spil som Boom Beach eller Clash of Clans (strategispil hvor man bygger en base, og samtidig angriber baser selv) hvor en notifikation sendes hvis noget er bygget færdigt i basen.

Den største fordel er at brugeren kan følge med i hvad der sker i appen. En anden fordel er at foruden apps vil styresystemet også sende notifikationer, eksempler: "battery is low", "your sound is muted".

Ulempen ved notifikationer er hvis der er mange af dem kan de virke spamvende, hvilket kan blive belastende for brugeren, også fordi han selv må fjerne dem hver gang.

## Forklar hvordan du har tænkt dig at bruge teknologien i din artikel

I min app vil jeg finde ud af hvordan man sender en notifikation efter 5 sekunder, og hvordan et ikon, en titel og indhold af tekst laves sammen med notifikationen. Også vil jeg gøre sådan at når man trykker på notifikationen vil man blive sendt til google på en browser. Foruden vil jeg også finde ud af hvordan man sender notifikationer mens appen er lukket ned.

## Forklar din løsning - hvad lavede du og hvorfor?

Til at starte med ville jeg finde ud af hvordan en notifikation bliver sendt på den mest simple måde. Efter at have ledt lidt omkring fik jeg konstateret at det skulle sættes op sådan:

- Få sat op en builder for notifikationen med typen 'NotificationCompat.Builder'.
- Ved brug af builderen, få sat en titel, indhold af tekst og et ikon.
- Få sat op en notifikation med typen 'Notification', og sæt den til at være et build af builderen.
- Få sat op en notificationmanager med typen 'NotificationManager' der henter et 'systemservice' som hedder NOTIFICATION\_SERVICE.
- Kald notificationmanager ved at hente notify, og sæt notifikationen og et id ind som parameter.
- Notifikationen burde nu komme frem på telefonen.

Billede med eksempel nedenfor:

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext(), channelId: "1"); //helps constructing the typical notification layouts
builder.setContentTitle("Simple Notification"); //sets the string to be shown as a title for the notification
builder.setContentText("Hello World!"); //sets the string to be shown as the context text for the notification
builder.setSmallIcon(R.drawable.ic_launcher_foreground); //sets an image as an icon for the notification
Notification notification = builder.build(); //we are done and we parse the build to be of the type 'Notification'
NotificationManager nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE); //has some tools for notifications, in this case we are looking for 'notify'
nm.notify(100, notification); //notify sends the notification
```

Builderen kan også andre ting som f.eks. at stile teksten eller tilføje en aktion (f.eks. svare på en besked i notifikationen) men for at gøre det simpelt vil jeg fokusere på titel, content-text, ikon og et intent som videresendes til google.com når man trykker på notifikationen.

Efter dette ville jeg finde ud af hvordan jeg kunne sende en notifikation efter 5 sekunder, hvilket jeg syntes var vigtigt fordi ofte bruges notifikationer som påmindelser, et eksempel er hvis brugeren gerne vil vide hvornår pileårnet er bygget færdigt i et spil-app.

Jeg fandt ud af 2 måder:

- Én simple måde, hvor der bruges en countdowntimer, som så starter et thread efter 5 sekunder der starter notifikationen (ulempen her er at appen skal køre i baggrunden ellers vil brugeren ikke få notifikationen).
- Én lidt mere avanceret måde, hvor der bruges en alarmmanager, der efter 7 sekunder skal vågne og få sendt en pending intent (som er en aktion der skal ske i fremtiden) som indeholder en klasse der udvider en broadcastreceiver. Klassen har jeg kaldt for NotificationPublisher. Notifikationen sendes også gennem intent så klassen kan få fat i den. Når 7 sekunder er gået vil klassen modtage dataet, og sammen med en notificationmanager, vil det nu blive muligt at køre notifikationen. Fordelen her er at notifikation **altid** vil komme frem, lige meget om brugeren har lukket appen ned eller ej.

Efter at løst begge punkter og alt virkede som det skulle, fandt jeg ud af at jeg bedre kunne lide løsning 2, den er ikke ligeså simpel men virker hver gang.

## Nogle billeder fra koden med forklaringer:

```
private PendingIntent pendingIntent;

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

    seconds_delay_btn = (Button) findViewById(R.id.after_5_seconds_button);
    send_while_closed_btn = (Button) findViewById(R.id.even_if_closed_button);
    simple_send_notification_btn = (Button) findViewById(R.id.send_now_button);

    status_textview = (TextView) findViewById(R.id.status_textview);
    Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.google.com")); //this intent opens the browser and then google.com
    pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0, intent, flags: 0); //specifies an action to take in the future.
```

De røde streger viser hvordan jeg fik lavet én pending intent (aktion i fremtiden) der vil få telefonen til at åbne browseren og gå ind i google.com. Først laver jeg intent med reglen ACTION\_VIEW og så også selv hjemmeside-adressen for google. Dernæst sætter jeg pending intent til at have sammenhæng med intent.

```
countDownTimer = new CountDownTimer( millisInFuture: 5000, countDownInterval: 1000) { //set countdown to max 5 seconds with a 1 second interval,
    // that means that each tick happens every 1 second.
    @Override
    public void onTick(long millisUntilFinished) {
        status_textview.setText("notification is being sent... (will take 5 seconds)"); //kind of lazy but each tick just set this waiting message (could use millisUntilFinished)
    }

    @Override
    public void onFinish() {
        status_textview.setText("notification sent!"); //when 5 seconds have passed, send this 'done' message.
    }
};
countDownTimer.start(); //start the count down.

Handler handler = new Handler(); //schedules messages and runnables to be executed at some point in the future
handler.postDelayed(() -> {
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(getApplicationContext(), channelId: "1"); //helps constructing the typical notification layouts
    mBuilder.setContentIntent(pendingIntent); //sets so that if user clicks on notification, the intent that goes to google.com is being started
    mBuilder.setSmallIcon(R.mipmap.ic_launcher); //sets an image as an icon for the notification
    mBuilder.setContentTitle("Delayed notification"); //sets the string to be shown as a title for the notification
    mBuilder.setContentText("Hello World after 5 seconds"); //sets the string to be shown as the context text for the notification
    Notification notification = mBuilder.build(); //we are done and we parse the build to be of the type 'Notification'

    NotificationManager nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE); //has some tools for notifications, in this case we are looking for 'notify'

    nm.notify( id: 2, notification); //notify sends the notification
}, delayMillis: 5000); //sets the rule: start thread after 5 seconds.
}
```

Koden udfører: 5 sekunder forsinkelse på notifikation, virker ikke når appen er lukket. Først laves en countdowntimer der ender efter 5 sekunder, hvor der kan ske noget hvert sekund (onTick). Hvert sekund sendes den samme besked til et textview om at brugeren må vente 5 sekunder, og når 5 sekunder er gået vil der stå at opgaven er fuldført. Hernæst laver jeg en handler (et værktøj til om noget skal køre i fremtiden), der laver et postdelay-thread med samme regel som countdown-timeren (vent i 5 sekunder), når nedtællingerne er færdige sendes notifikationen.

## Hallur við Neyst

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext(), channelId: "3"); //helps constructing the typical notification layouts
builder.setContentIntent(pendingIntent); //sets so that if user clicks on notification, the intent that goes to google.com is being started
builder.setTitle("Notification When App Is Closed"); //sets the string to be shown as a title for the notification
builder.setContentText("Hello, your app is not open!"); //sets the string to be shown as the context text for the notification
builder.setSmallIcon(R.mipmap.ic_launcher); //sets an image as an icon for the notification
Notification notification = builder.build(); //we are done and we parse the build to be of the type 'Notification'

Intent notificationIntent = new Intent( packageContext: this, NotificationPublisher.class); //this intent uses the NotificationPublisher class to notify on receive
notificationIntent.putExtra(NotificationPublisher.NOTIFICATION, notification); //send data from MainActivity with the intent, this case the notification
PendingIntent pendingIntent2 = PendingIntent.getBroadcast( context: this, requestCode: 0, notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT); //specifies an action to take

long futureInMillis = 7000L; //set the rules here. Currently we want to wait 7 seconds until notification is being sent
AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE); //get the alarm manager
alarmManager.set(AlarmManager.ELAPSED_REALTIME_WAKEUP, futureInMillis, pendingIntent2); //wake it up after 7 seconds! then do pendingIntent2!
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hvnl5.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Notification Demo App"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".NotificationPublisher" />
    </application>
</manifest>
```

```
public class NotificationPublisher extends BroadcastReceiver {
    public static String NOTIFICATION = "notification";

    @Override
    public void onReceive(Context context, Intent intent) {
        NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);

        Notification notification = intent.getParcelableExtra(NOTIFICATION); //get the notification
        notificationManager.notify( 0, 3, notification); //send the notification
    }
}
```

Koden udfører: 7 sekunders forsinkelse på notifikation, virker også når appen er lukket ned.

På det første billede laves der først en notifikation. Dernæst laves der en intent med sammenhæng for klassen NotificationPublisher og notifikationen bliver sendt gennem den. Herefter laves en pending intent med sammenhæng for intent. Til sidst er der en alarmmanager der med regelen at efter 7 sekunder så kører vi pending intent.

På det næste billede er vi inde i manifest-filen, det er vigtigt at tilføje receiver tagget for klassen NotificationPublisher ellers vil klassen ikke kunne modtage noget.

På det tredje billede udvides klassen med BroadcastReceiver som vil kræve at man skal bruge en onReceive metode. Denne metode er hvad vil køre efter 7 sekunder, i vores tilføjelse vil vi gerne sende en notifikation så jeg henter en notificationmanager, henter også notifikationen vi fik sendt fra MainActivity og til sidst starter jeg den med notificationmanageren.

### **Vurder din løsning – hvordan gik det?**

Jeg er tilfreds med min løsning. Der var en fejl angående NotificationCompat builderen i logcat, men siden appen virkede som jeg ville, lod jeg det ligge. Jeg er glad for at komme gennem hvad jeg ville klare, dog kunne koden sættes bedre op og layout ikke er særligt pænt

### **Konklusion – Foreslår du din løsning og hvad lærte du**

Hvis at en app-udvikler har tænkt sig at bruge notifikationer på den normale måde (der er andre måder, f.eks. med firebase) ville jeg anbefale mit projekt. Grunden er at projektet både viser hvordan en simpel notifikation bliver sendt (hvilket er det første man ville undersøge), men også hvordan man sender notifikationen efter en bestemt tid (hvilket jeg ville argumentere for udviklere i mange tilfælde også ville undersøge)

Jeg fik lært om hvordan en notifikation sættes op, hvad den kan indeholde, og hvordan man laver en klik-opførsel på den. Jeg fik også lært forskellige måder for nedtælling for hvornår notifikationen sendes, både mens man er i appen og mens man er ude af appen.