

|  |
| --- |
|  |
| Audio & Visual productions |
| Documentatie: Infinity Mirror |
|  |
|  |

# Doelstelling

Het project dat wij hadden gekozen was de ‘Infinity Mirror: salontafel’. Dit is een spiegel die zich baseert op een optische illusie.

Door gebruik te maken van 1 gewone spiegel (onderaan), 1 half doorlatende spiegel (bovenaan) en een LED‑strip tussenin lijkt het alsof je door de salontafel kunt kruipen. Omwille van het feit dat de bovenste spiegel ongeveer 50% van het licht reflecteert, zal een deel terugkaatsen richting de gewone spiegel en de overige lichtstralen zullen op ons netvlies terugkaatsen.

Om deze reden kan de LED‑strip zich verscheidene keren herhalen en lijkt het alsof deze ‘oneindig’ herhaald wordt.

Onze opdracht was om dergelijke opstelling te bouwen. Het was echter niet voldoende om een normaal werkende infinity mirror te maken. We moesten ervoor zorgen dat deze op een interactieve manier werkte met omstaande mensen.

Een voorbeeld was om mensen de salontafel te laten besturen en zelf hun effecten te laten kiezen. Dingen zoals het tracken van een glas dat op de tafel stond en een ingebouwde equalizer behoorden ook tot het interactief spectrum.

# Installatie instructies

## Server – Infinity Mirror

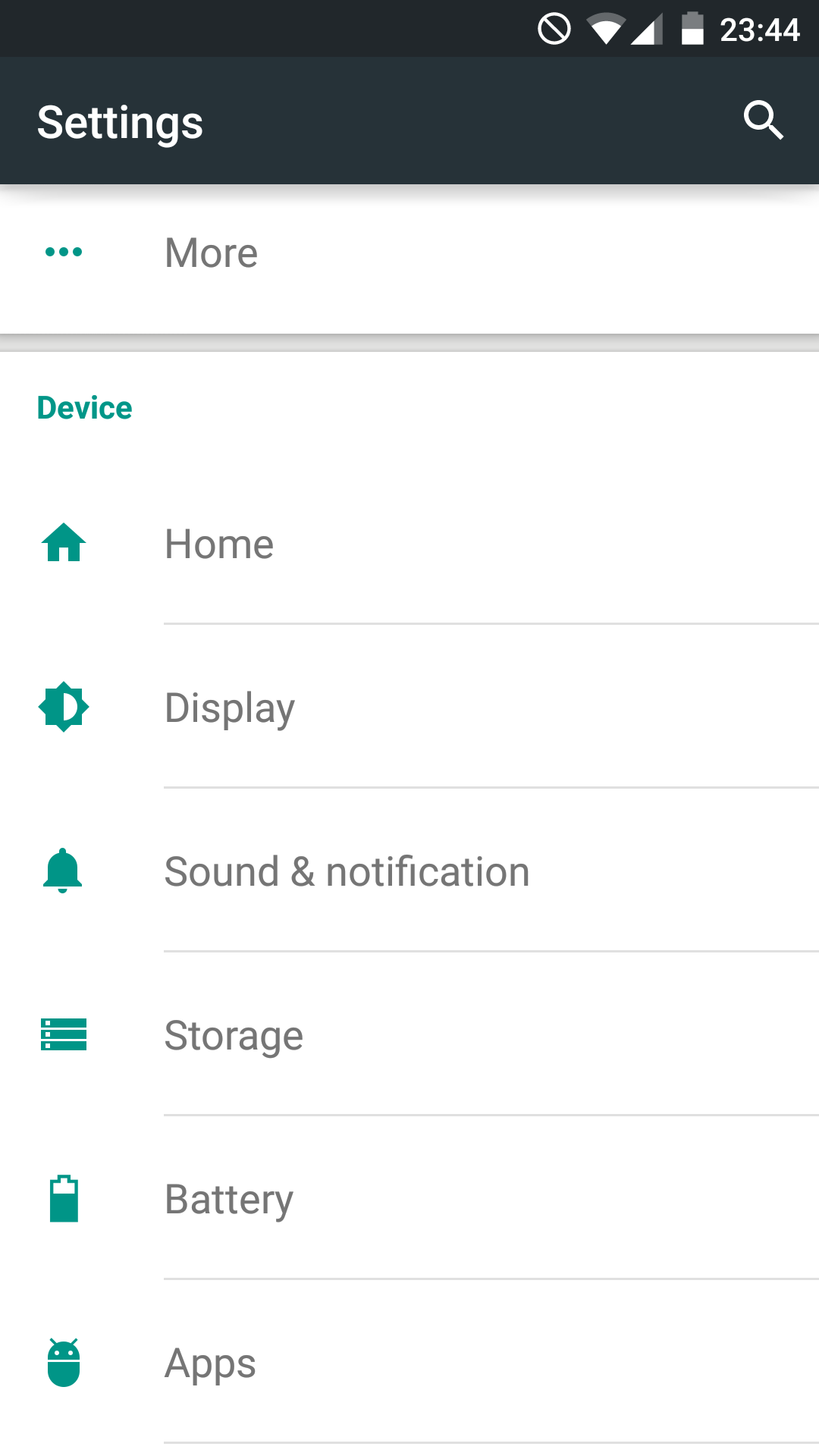
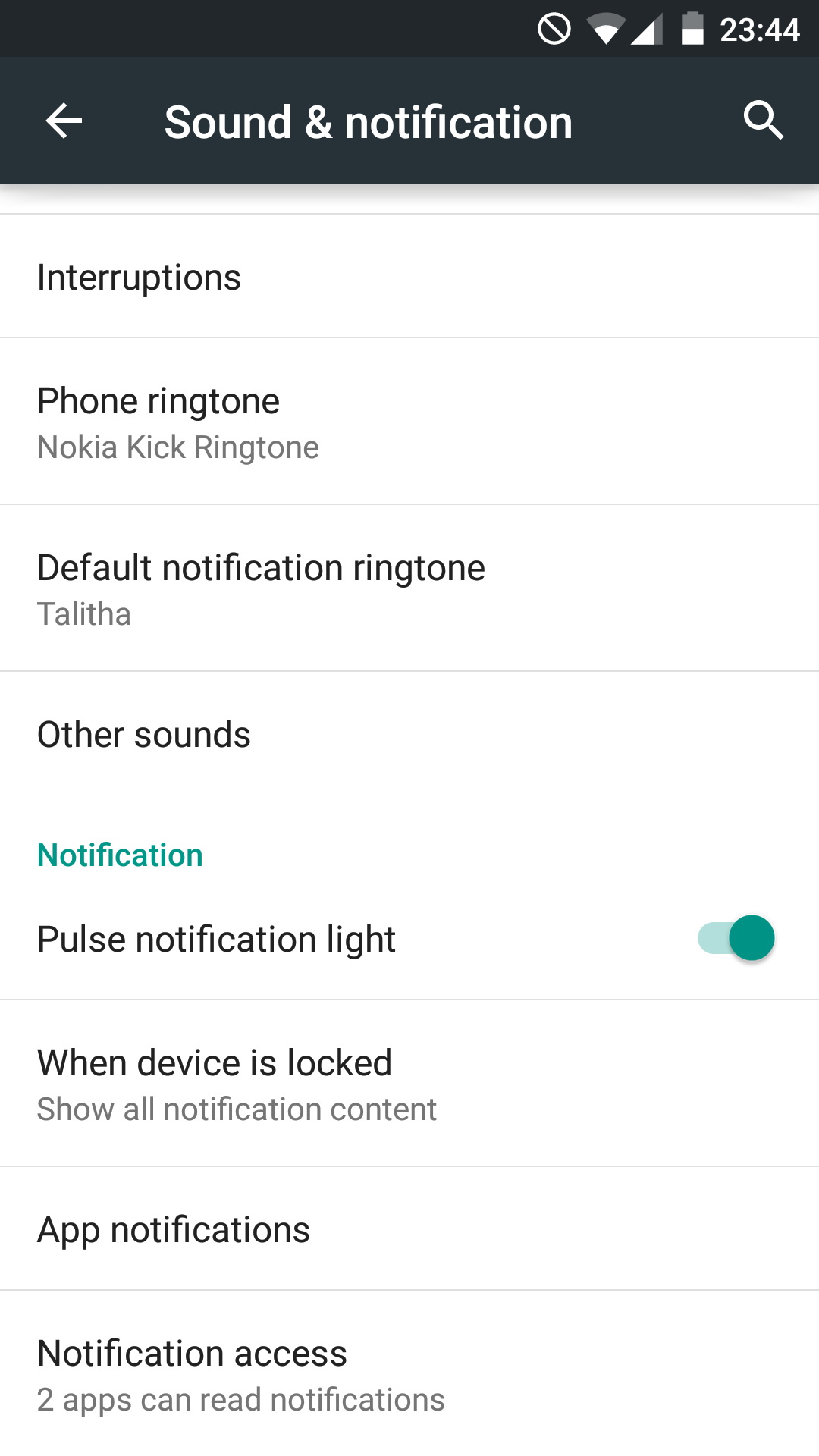
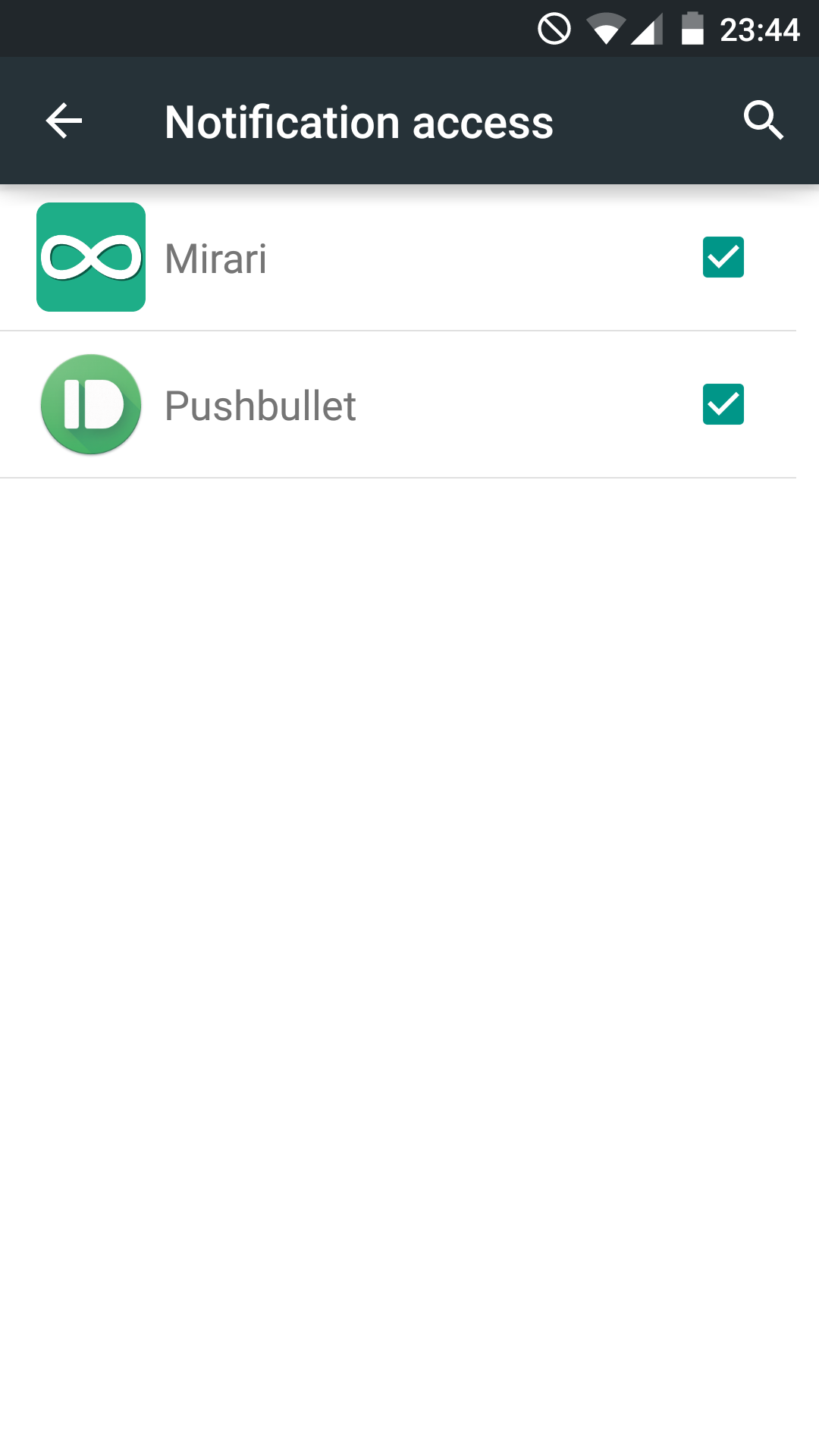
Men steekt de tafel in het stopcontact en sluit de netwerkkabel aan op het netwerk. Heden moet de node server nog gestart worden via een ssh terminal zoals putty. Het uit te voeren commando is “sudo node mirrorapi/apigrid.js” en “sudo node mirrorapi/apistrip.js”. Nu bent u klaar.

## Client – Android applicatie

De app kan geïnstalleerd worden via de Play Store op Android devices met een besturingssysteem hoger dan 4.3 Jelly Bean.

Download URL: <https://play.google.com/store/apps/details?id=be.howest.nmct.android>

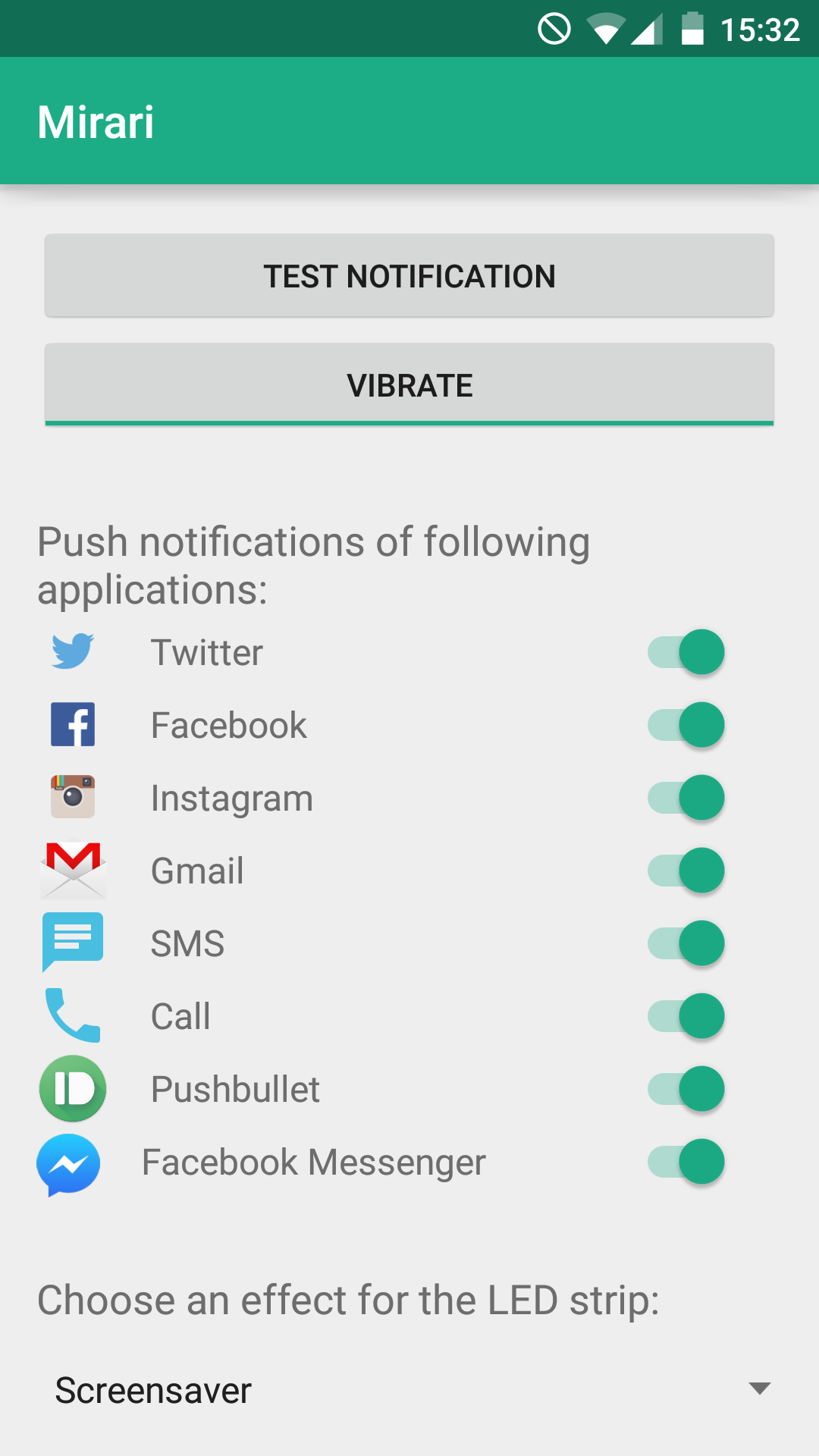
Het gebruik is redelijk eenvoudig. U dient gewoon de applicatie te installeren. Eenmaal geïnstalleerd gaat u naar ‘Instellingen’ > ‘Sound & notification’ > ‘Notification access’ > ‘Mirari’ aanvinken.

*Afbeelding: toestaan van notificatie toegang*

Nadien kunt u de applicatie opstarten en via de ‘Test notification’ knop controlleren of uw telefoontoestel verbinding kan maken met de spiegel. Normaal zou het middelste grid een willekeurige kleur hebben met een witte horizontale streep erdoor.

Via handige switches kunt u zelf kiezen van welke app notificaties getoond moeten worden op de spiegel.



*Afbeelding: screenshot van de applicatie*

# Werking

## Server – Infinity Mirror

De spiegel maakt gebruik van een Raspberry Pi die verbonden in met een LED‑strip en een zelfgemaakte LED-matrix. De LED-strip werkt volgens het WS2801 protocol. De LED-matrix volgens het WS2811 protocol. Er zijn ook nog 4 servo motors aanwezig om de spiegel lichtjes te doen trillen of bewegen.

Op de Raspberry Pi draait er een Node.js server die alle licht effecten bestuurd/controleert. Een REST API zorgt op zijn beurt dan weer dat de Android applicatie via het internet kan communiceren met de server.

Onze API vangt de requests op en voegt de notificatie toe aan een array en toont deze in de spiegel. Zodra er meerdere notificaties zijn worden deze om de 2 seconden getoond in het middelste grid. Door middel van handig te herkennen iconen weet de gebruiker meteen van welke applicatie hij/zij een notificatie heeft op zijn telefoon.

Via de instellingen is het ook mogelijk de spiegel te laten trillen telkens er een melding binnenkomt.

Door simpel weg de notificatie weg te swipen verdwijnt deze op de spiegel. Indien de gebruiker gebeld wordt zal deze meteen verschijnen.

## Client – Android applicatie

Op de client draait er een service die constant checkt of er nieuwe notificaties binnen komen. Eenmaal een bepaalde packagename van een binnengekomen melding overeenkomt met een toegestane app zal er een GET gestuurd worden naar de Raspberry Pi. Via URL queries kunnen we aan de server meedelen om welke package het gaat zodat deze het juiste icoon kan tonen.

Hetzelfde geld ook voor het wegswipen van een notificatie. De service detecteert dit en stuurt terug een commando naar de Pi om de melding te verwijderen.

In een preference XML file worden de voorkeuren van de gebruiker bijgehouden zodat deze niet iedere keer opnieuw ingesteld moeten worden.

# Aanpak en realisatie

We zijn begonnen met het maken van een houten frame waarin dan een LED‑strip gekleefd is. Omdat we enige vorm van beweging wilden creëren in de spiegel zijn we op zoek gegaan naar een buigbare spiegel. Het eerste idee was een polystyreen spiegel, maar na enig opzoekingswerk zijn we tot de conclusie gekomen dat er zich plooien zullen vormen na enkele keren te buigen.

Bij een bezoekje aan de plaatselijke doe‑het‑zelf winkel vonden we een reflecterende folie die gebruikt wordt als zelfklevende spiegel. Er was echter niet de gewenste reflectie die nodig was om de optische illusie te kunnen creëren.

Met hulp van een IPO‑student zijn we bij een bedrijf geweest dat gespecialiseerd is in allerlei soorten plexiglas. Zij verkochten spiegels die redelijk flexibel waren. We besloten dat dit waarschijnlijk de beste optie zou zijn en hebben er eentje besteld.

De half reflecterende spiegel hebben we van een bedrijf uit Ieper kunnen krijgen. Zij maken ramen met spiegelglas. Na het vertellen van ons project hadden ze een perfect stuk glas op overschot dat we mochten gebruiken.

Hierna zijn we begonnen met de opstelling in elkaar te steken en eens te testen of alles goed werkte. We zagen meteen dat alles goed zat en het gewenste resultaat bereikt kon worden.

Op dit moment werd de LED‑strip nog via een Arduino aangestuurd.

Omdat er dus een LED‑grid en 4 servo’s bijkwamen zijn we overgeschakeld naar een Raspberry Pi. Het was immers makkelijker omdat een Raspberry meerdere threads aankan en de draadloze communicatie tussen server en client hoefde niet meer te gebeuren via NRF24l01 RF‑modules. Alles kon meteen via het net aangestuurd worden en een Pi zou het verdere verloop van het project aangenamer maken.

We hebben dus alles herschreven zodat de aansturing van alle LEDs werkt via de Raspberry Pi en hebben ook de API geschreven om te kunnen communiceren. De overschakeling van Arduino naar Pi verliep redelijk vlot. Na een halve dag ‘kennismaken’ konden we zonder problemen alles werkende krijgen.

Terwijl één iemand bezig was met de afwerking van spiegel was iemand anders bezig met het schrijven van de Android applicatie.

Na enkele keren grondig te testen konden we beginnen met het toevoegen van extra features zoals support voor meerdere applicaties, het kiezen van verschillende effecten, meldingen in een carrousel laten draaien, …

# Problemen en oplossingen

Gedurende de volledige loop van dit project waren er niet zoveel vernoemings waardige problemen. Een van de grootste probleem dat we hadden was ervoor zorgen dat onze Node.js server meerdere LEDs kon aansturen en constant te laten loopen.

Een oplossing hiervoor was zorgen voor een oneindige loop die onderbroken wordt wanneer men een nieuw effect heeft gekozen.

De servo's gebruiken eenzelfde pin als de LED-matrix we hebben dit proberen te verhelpen maar de pin wordt niet vrijgegeven als men de servolibrary benut.

# Ideeën naar mogelijke uitbreidingen

Het model dat we nu hebben is nog eerder een prototype. De afwerking en materiaal keuze leent er zich niet toe om het product te verkopen.

De gehele opstelling kan ook nog iets dunner gemaakt worden dan dat het nu is.

De applicatie runt nu enkel voor onze spiegel. Een optie om het IP‑adres van de tafel in te stellen in de Android app zou er nog bij kunnen komen. Op deze manier kan de tafel op ieder netwerk aangesloten worden en zonder veel problemen bestuurd worden.

Zoals gezegd hebben we enkel een Android applicatie. Maar dankzij het gebruik van de Raspberry Pi in combinatie met een Node.js server en API maakt het mogelijk om via een web platform de spiegel over het internet te laten besturen. We zouden dus een website kunnen maken zodat je vanuit de browser je spiegel thuis kunt bedienen.

Een laatste uitbreiding is ervoor zorgen dat de gebruiker zelf figuren kan tekenen in de LED‑matrix, effecten maken en ze meteen ook opslaan.

# Referenties en bibliografie