Deze apps worden gemaakt uit het perspectief van iemand die reeds de basis van native Android, native iOS, React Native, Xamarin Forms en Flutter beheerst.

Het ontwerp is gemaakt met Adobe XD en gebaseerd op hoe de Android variant van de app er uit zal zien.

Daarom zal eerst de Android app gemaakt worden. Deze app wordt gemaakt met de programmeertaal Kotlin. De documentatie over Android en de meeste gebruikte packages is te vinden om <https://developer.android.com>.

Om deze app eenvoudig te houden, zal niet gebruik worden gemaakt van de MVVM-architectuur die door Google aanbevolen wordt. Wel zal er slechts gemaakt worden van één activity en meerdere fragments, wat volgens Google een best practice is. Ook wordt er gebruik gemaakt van de Navigation Component om de navigatie te vereenvoudigen. Dit is een nieuwe component die deel uitmaak van de Android Jetpack library.

Het aanmaken van het project ging vrij eenvoudig via een wizard.

Bij het maken van de layout voor de detail pagina dook een eerste probleem op, bij het zoeken naar documentatie over constraints om precies te zijn. Er is een verschil tussen Android en AndroidX packages. AndroidX is het open-source project dat het Androidteam gebruikt voor het ontwikkelen, testen en uitbrengen van libraries in Jetpack. Als je bijvoorbeeld documentatie opzoekt over ConstraintLayout of BottomNavigationView, zal je bronnen uitkomen die verschillende packages beschrijven. Dit kan voor verwarring zorgen, zeker bij beginnende programmeurs.

Er zijn nog een aantal kleine zaken die opvallen. Eenvoudige zaken als de kleur van de icoontjes in de toolbar is niet zo eenvoudig als je zou denken. In plaats van de kleur rechtstreeks in de XML van de toolbar aan te passen, moet je er een nieuwe stijl aan toekennen. Deze stijl bevat een attribuut voor het meegeven van de gewenste kleur voor de icoontjes in de toolbar.

Nog iets dat opvalt zijn de developer tools. Zoals later wel besproken zal worden, hebben Flutter en React Native Hot Reload. Dit zorgt ervoor dat je je wijzigingen sneller kunt waarnemen. Bij native Android kan je wel ‘Apply changes’ doen zodat de app niet volledig opnieuw opgestart moet worden, maar er moet wel eerst een gradle build gebeuren. Dit neemt veel tijd in beslag, zeker tegenover Hot Reload.

Bij Android is er standaard geen functionaliteit om afbeeldingen via een url op te halen. Daarvoor dienen 3rd party packages als Picasso of Glide voor gebruikt worden. In dit voorbeeld wordt gebruik gemaakt van Glide.

Een grote moeilijkheid was het programmatisch toevoegen van rijen aan de tabel op de detailpagina. In de documentatie van Android stond geen voorbeeld, en daarom werden meerdere StackOverflow pagina’s geraadpleegd. Uiteindelijk werd een oplossing gevonden, maar deze lijkt vrij omslachtig.

Ook is bij het grid niet eenvoudig om een border in te stellen. Er moet eerst een aparte XML file gemaakt worden waarin een strook gedefinieerd wordt. Dit moet dan als achtergrond ingesteld worden.