
◆ Mobile Devices

➤ *Welke soorten mobile devices zijn er?*

- Goed begin, maar breid het uit met gebruikstoepassingen.

Voorbeeld van verdieping:

Smartphones – Handzame apparaten die geschikt zijn voor bellen, browsen en het draaien van apps. Ze zijn het primaire device voor communicatie onderweg.

Tablets – Groter scherm, ideaal voor multimedia en presentaties. Wordt veel gebruikt in onderwijs en zakelijke meetings.

Laptops – Meer rekenkracht, toetsenbord en grotere schermen. Geschikt voor softwareontwikkeling, grafisch werk of productiviteit.

Smartwatches – Wearables die data meten zoals hartslag, notificaties tonen en functioneren als verlengstuk van de smartphone.

Verbinding met eindproduct: "Bij het ontwikkelen van een mobiele app moet men rekening houden met het feit dat de gebruikers verschillende devices gebruiken, wat gevolgen heeft voor UI-responsiviteit en performance."

➤ *Welke besturingssystemen bestaan er en wat is het marktaandeel?*

- Voeg impact toe van OS-keuze op app-ontwikkeling.

Voorbeeld van verdieping:

Android (71,5%) – Open source, grote fragmentatie, vereist veel testen op verschillende toestellen.

iOS (28%) – Gesloten ecosysteem, consistente hardware, maar hogere eisen bij App Store goedkeuring.

Windows Phone (<0,1%) & BlackBerry OS (0%) – Verouderd, nauwelijks nog ondersteund. Niet relevant bij moderne appontwikkeling.

Verbinding met eindproduct: "De keuze om een native app voor Android of iOS te ontwikkelen hangt sterk af van de doelgroep. Voor een brede internationale gebruikersgroep is Android de logische keuze, maar iOS wordt vaak gekozen voor premium apps vanwege de hogere opbrengst per gebruiker."

➤ ***Wat zijn de meest bekende producenten van mobile devices?***

- Voeg regio en doelgroep toe.

Uitbreiding:

Samsung (20%) – Breed aanbod van low- tot high-end. Populair in Europa en Azië.

Apple (19%) – Premium markt. Sterk in de VS, Japan en West-Europa.

Xiaomi (14%) – Betaalbare high-spec toestellen. Populair in India en Zuid-Amerika.

Vivo (8%) – Gericht op Aziatische markten. Sterk op camerakwaliteit.

Verbinding met eindproduct: "Bij performance testing moet rekening gehouden worden met veelgebruikte toestellen van deze merken."

➤ ***Typische kenmerken van mobile devices***

- Voeg toe waarom deze kenmerken belangrijk zijn voor ontwikkelaars.

Uitbreiding:

Schermmresolutie – Belangrijk voor layout/UI design. Responsive design is essentieel.

Opslagcapaciteit & RAM – Bepalen of de app vlot draait of opslagproblemen oplevert.

Batterijcapaciteit – Apps moeten energiezuinig zijn; background processing minimaliseren.

Processor – Belangrijk bij apps met zware grafische eisen (zoals games).

➤ ***Typische functies van mobile devices***

- **Verbind functies met app-functionaliteiten.**

Voorbeeld van verdieping:

Foto's maken – Essentieel voor social media apps (Instagram).

Navigatie/GPS – Belangrijk voor bezorg- en locatiegebaseerde apps (Uber).

Beveiliging – Noodzakelijk voor banking apps, implementatie via biometrische authenticatie.

♦ **Mobile Apps**

➤ ***Technische types apps (native, web, hybrid)***

- **Inhoud is correct. Voeg architecturale verschillen en use cases toe.**

Voorbeeld aanvulling:

Native – Gebouwd in platform-eigen taal (Swift voor iOS, Java/Kotlin voor Android). Beste voor apps met complexe interacties of hardwaretoegang.

Web apps – HTML5, draaien in browser. Goed voor eenvoudige content-gebaseerde apps.

Hybrid – Gebruik van frameworks als Ionic of React Native. Sneller ontwikkelen, maar soms minder performant.

➤ ***Voor- en nadelen per type***

- **Goed gestructureerd. Voeg praktijkvoorbeelden toe.**

Uitbreiding:

Native nadeel: De ABN AMRO app moet apart onderhouden worden voor Android en iOS.

Web app nadeel: Bol.com webapp kan geen pushnotificaties sturen op alle apparaten.

Hybrid voorbeeld: De eerste versie van Uber was hybride, maar is overgestapt naar native vanwege performance.

◆ Mobile Application Development

➤ *Wat is een IDE?*

- Definitie is kort. Voeg concreet voorbeeld toe.

Uitbreiding:

Een IDE (zoals Android Studio) biedt een grafische interface om code te schrijven, fouten op te sporen en simulators te gebruiken om apps te testen. Het versnelt de ontwikkelcyclus.

➤ *Welke IDE's zijn geschikt?*

- Voeg per IDE toe welk platform ermee wordt ondersteund.

Voorbeeld:

Android Studio – Voor Android apps met Java/Kotlin.

Xcode – Voor iOS/macOS apps met Swift of Objective-C.

Visual Studio – Voor multiplatform development (bijv. met Xamarin of MAUI).

➤ *Programmeertalen voor mobile apps*

- Voeg toe welke taal bij welk platform hoort.

Voorbeeld:

Java – Veel gebruikt voor Android.

Swift – Taal van Apple voor iOS/macOS.

JavaScript – Gebruikt in hybride apps via frameworks zoals React Native of Ionic.



Eindconclusie / Verwijzing naar eindproduct

“Bij het ontwikkelen van onze eigen mobile app (eindproduct) is het van belang dat we begrijpen welk type app we maken, voor welk platform, en wat de technische implicaties zijn. Door gebruik te maken van de juiste IDE en programmeertaal kunnen we inspelen op de meest gebruikte mobile devices en hun specificaties. Dit bevordert compatibiliteit en gebruikerservaring.”



Samenvattend advies

Punt	Huidige status	Aanbeveling
Inhoud	Correct maar oppervlakkig	Meer context, praktijkvoorbeelden
Verbinding met eindproduct	Ontbreekt	Voeg dit consistent toe
Bronnengebruik	Aanwezig	Meer diversiteit en betrouwbaarheid gewenst (bijv. Statista, Gartner)
Structuur	Logisch	Handhaaf dit, maar werk per onderdeel dieper uit
