

Oplevering

Future media platforms

PSV Eindhoven



Date	:	14/02/2022
Author	:	Dary van Sleuwen, Rick van Wijk, Kimberly Huggers, Sem Sodermans, Max Kroeze

Inhoudsopgave

Inhoudsopgave	2
Technologie - best practices & advies	3
Hybrid Frameworks	3
Flutter	3
Frontend info & practices	3
Bekijk de frontend repo op Github voor voorbeelden	3
Backend info & practices	3
Bekijk de backend repo op Github voor voorbeelden	4
Notificaties info & practices	5
Prototypes/ontwerpen	5
Onderzoeken	7
Conclusie, advies & vervolgstappen	9

Technologie - best practices & advies

Hybrid Frameworks

Als we kijken naar welke frameworks en beschikbaar zijn en wat deze bieden qua voor- en nadelen performance en UX/Developer Experience kunnen we concluderen dat Flutter en React Native de beste keuzes zijn om een soepele app te maken die volledig native is waarbij native features zoals NFC, camera, GPS etc. snel en gemakkelijk te gebruiken zijn.

Flutter

Frontend info & practices

Flutter is dus een goed ontwikkeld framework om hybrid apps mee te maken. Het maakt gebruik van Widgets en je kan deze zelf ook custom maken. Gelukkig zijn er al veel ingebouwde Flutter widgets die je direct kan gebruiken. Flutter maakt gebruik van hot reloading tijdens het developen van een app. Maak hier dus goed gebruik van.

Een nadeel van de widget en de ui van je app is dat je veel moet nesten in de code. Hierdoor kan je code onoverzichtelijk worden. Het is daarom belangrijk dat je de widget opsplits in kleine UI stukjes. Je kan bijvoorbeeld in een apart bestand zo widget maken en deze bestanden weer te importeren in een pagina. Zo heb je een algemene bestand per pagina met daarin alle componenten die zijn ingeladen. Zo heb je grote onderdelen van je UI's op een aparte plek staan en is het handiger om daarna terug te kijken in de structuur van de UI.

Om het probleem met de widget nesting tegen te gaan kan je verschillende d2ev tools installeren om je te helpen met versnellen van je development process. Zo'n tools automatiseren een heleboel zoals een lange widget verplaatsen in een andere widget. Zo hoeft je niet een hele rits aan code te kopiëren en te hopen dat je alles goed hebt gekopieerd.

Resources

Bekijk de **frontend** repo op [Github](#) voor voorbeelden

Backend info & practices

Voor het regelen van bestellingen en de communicatie tussen de app en het admin panel wordt aangeraden om een live verbinding te creëren met de technologie **Socket IO**. Hierdoor kan er gemakkelijk een nieuwe bestelling worden aangemaakt in de app en doorgestuurd worden naar de backend door middel van zogenoemde socket events. Met een andere event kun je de

nieuwe bestelling weer meteen doorsturen naar de admin panel. Dit zijn HTML, CSS JavaScript applicaties die alle bestellingen in het stadion ontvangen en afhandelen. Via dit panel kan iemand achter de balie met deze applicatie bestellingen op “**klaar**” zetten zodat een gebruiker via zijn telefoon weer op de hoogte wordt gesteld van zijn bestellingen die klaar is.

Ook kun je via dit panel de unieke QR code van een bestellingen scannen. Als je scan succesvol is, krijgt de gebruiker zijn eten of drinken en wordt de bestellingen gearchiveerd. Vervolgens wordt er weer een socket event afgevuurd om de app van de gebruiker op te hoogte te stellen dat de bestellingen is gescand en ontvangen.

Je moet bij Socket IO wel opletten dat je aangemaakte events goed gestructureerd zijn in de code. Als je veel events heb, kan het onoverzichtelijk worden en dat gaat ten kosten van de kwaliteit van de code en dus van de app

Resources

Bekijk de **backend** repo op [Github](#) voor voorbeelden

Notificaties info & practices

Soorten notificaties

Flutter biedt support voor twee soorten notificaties aan; push notifications (*Notifications: Flutterfire*) & local notifications (*Flutter_local_notifications: Flutter Package 2022*).

“Local notifications originate from the application itself, as opposed to Push notifications which are triggered from a remote server.” (Tomerpacific, 2021)

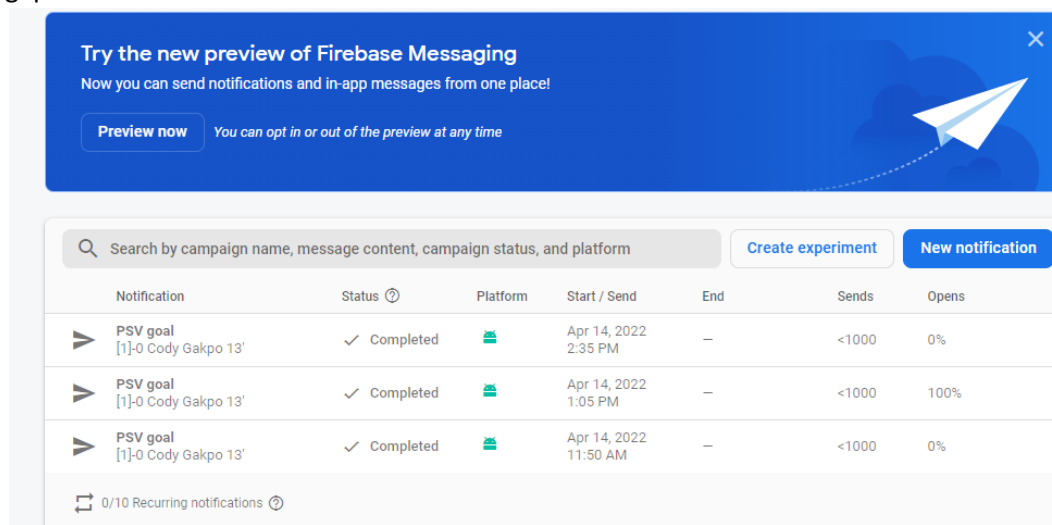
Flutter push notifications support

- Flutter wordt ondersteund door Firebase Cloud Messaging.
- Firebase ondersteund iOS, Android, JavaScript en meer talen (*firebase documentation*).
- Firebase Cloud Messaging laat je ook custom berichten sturen vanaf je [eigen servers](#).
- Third-party SDK's zoals OneSignal (*App.onesignal.com*) bieden ook Flutter support.

Local notifications zijn handig om te gebruiken wanneer de gebruiker de app of de foreground heeft geopend. Notificaties zullen dan nog wel binnenkomen in tegenstelling tot push notificaties (met Cloud Messaging).

Wat zijn de bijbehorende best practices voor de Flutter notificatie functionaliteiten?

- Local notifications zijn het beste om te gebruiken wanneer de gebruiker de app op de foreground heeft geopend. Notificaties zullen dan nog wel binnenkomen in tegenstelling tot push notificaties (met Cloud Messaging).
- Voor de beste push notificatie support met Flutter, wordt er aangeraden [Firebase](#) te gebruiken. Firebase biedt een eigen dashboard/omgeving aan waarin notificaties ontwikkelt en gemanaged kunnen worden. Firebase geeft je bijvoorbeeld de optie om gebruikers te segmenteren en dus gepersonaliseerde notificaties te sturen.



- Gebruik geen GIF's of video bestanden voor notificaties. Deze worden niet ondersteund door Flutter. Alleen de typische statische bestanden JPG/JPEG, PNG, ICO, etc. worden ondersteund.

Resources

Bekijk de repo op [Github](#) voor voorbeelden

Prototypes/ontwerpen

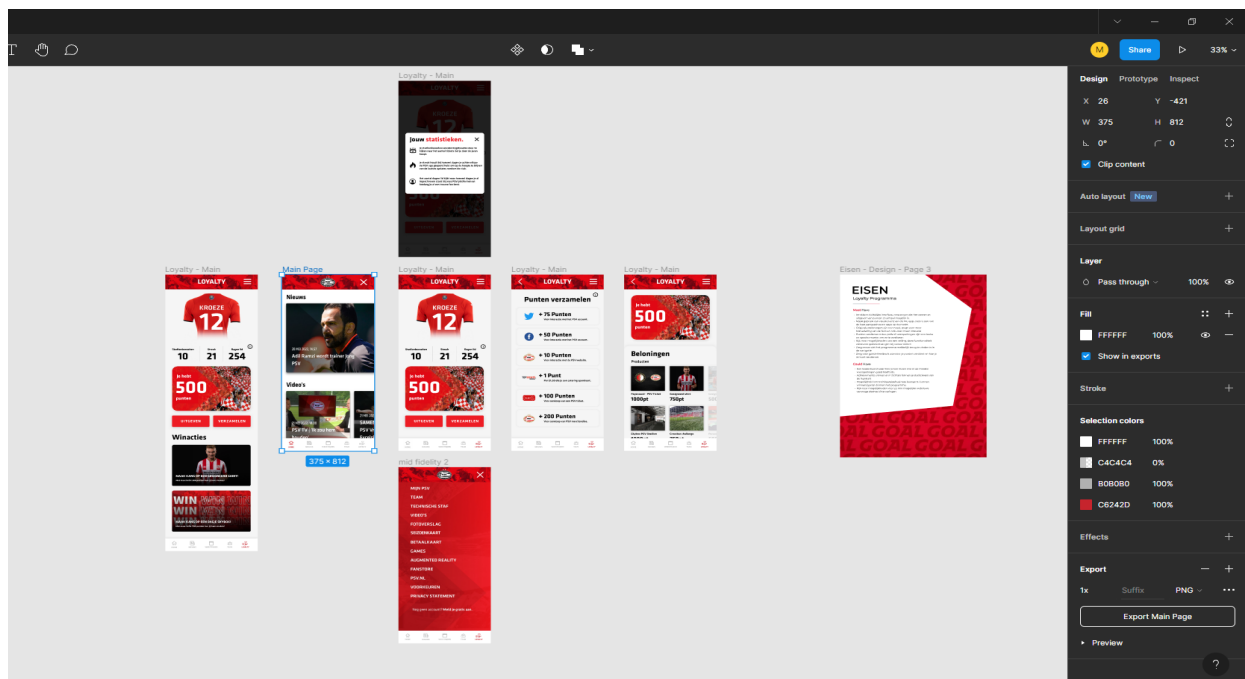
We hebben het project opgedeeld in 2 projecten om zo voor zowel media als voor PSV aan de eisen te kunnen voldoen. Voor ons media project zijn we aan de slag gegaan met het uitwerken van enkele functies voor de PSV app en deze zijn we gaan omzetten in een Hi-Fi prototype, hiervoor hebben we gebruik gemaakt van de designtool Figma. Uiteindelijk zijn we aan de slag gegaan met onderstaande features:

- Seizoenkaart
- Notificaties
- Loyaliteitsprogramma
- Tickets
- Catering

Om de gemaakte designs te kunnen bekijken kun je op onderstaande link klikken en wordt je naar de desbetreffende **Figma** gebracht waarin je alles naar wens kunt aanpassen indien nodig.

- <https://www.figma.com/file/gzKnndA1tw3dpHIGK0ff1L/Designs-%26-Prototypes?node-id=151%3A2>

Ook is het mogelijk om de verschillende schermen te exporteren, dit kan gemakkelijk met het menu dat zich aan de rechterzijde bevindt. Je kunt kiezen uit: export als pdf, png, jpg & svg (je moet eerst op een frame klikken en daarna krijg je deze opties).



Onderzoeken

Algemeen

Projectplan

- *Semester 6 - PSV // algemeen*

Doelgroep

- *Semester 6 - PSV // algemeen*

Interviews

Interviews conclusie

- *Semester 6 - PSV // interviews*

Interviews vragen

- *Semester 6 - PSV // interviews*

1e set interviews

- *Semester 6 - PSV // interviews*

2e set interviews

- *Semester 6 - PSV // interviews*

3e set interviews

- *Semester 6 - PSV // interviews*

Oriënterend onderzoek

Hybrid Framework Onderzoek

- *Semester 6 - PSV // oriënterend onderzoek*

Huidige ICT Architectuur

- *Semester 6 - PSV // oriënterend onderzoek*

Concurrenten Frameworks

- *Semester 6 - PSV // oriënterend onderzoek*

Seizoenkaart

Concurrenten Analyse

- *Semester 6 - PSV // literatuuronderzoek // seizoenskaartsysteem*

Mobiele Trend Onderzoek

- *Semester 6 - PSV // literatuuronderzoek // seizoenskaartsysteem*

Loyaliteitsprogramma

A/B Testing

- *Semester 6 - PSV // literatuuronderzoek // loyaliteitssysteem*

Survey Research

- *Semester 6 - PSV // literatuuronderzoek // loyaliteitssysteem*

Concurrenten Analyse

- *Semester 6 - PSV // literatuuronderzoek // loyaliteitssysteem*

Tickets

AB testing

- *Semester 6 - PSV // literatuuronderzoek // ticketssysteem // tests*

Thinking Out Loud

- *Semester 6 - PSV // literatuuronderzoek // ticketssysteem // tests*

Concurrentie Analyse

- *Semester 6 - PSV // literatuuronderzoek // ticketssysteem*

Huidige situatie ticketssysteem

- *Semester 6 - PSV // literatuuronderzoek // ticketssysteem*

Ticket scannen manieren

- *Semester 6 - PSV // literatuuronderzoek // ticketssysteem*

Voetbaltickets verkoop

- *Semester 6 - PSV // literatuuronderzoek // ticketssysteem*

Catering

Catering Onderzoek

- *Semester 6 - PSV // literatuuronderzoek // cateringsysteem*

Notificaties

Trendanalyse notificaties

- *Semester 6 - PSV // literatuuronderzoek // notificatiesysteem*

Huidig notificatiesysteem

- *Semester 6 - PSV // literatuuronderzoek // notificatiesysteem*

Flutter Notificaties Onderzoek

- *Semester 6 - PSV // literatuuronderzoek // notificatiesysteem*

Conclusie, advies & vervolgstappen

Techniek

Flutter is geschikt om een goedkope(re), onderhoudbare cross-platform app te ontwikkelen die alle huidige en toekomstige features van PSV bevat.

Als vervolgstap raden we aan diepgaand onderzoek te doen naar hoe Flutter met de huidige ICT-architectuur van PSV samen kan werken. Dus, hoe zou een nieuwe PSV app gemaakt met Flutter gebruik kunnen maken van Gracenote, Sports Alliance en XperienceCentral. Wij raden aan dit onderzoek door ICT & Software studenten te laten uitvoeren, want zij hebben meer kennis in het onderzoeken van technologieën en bovendien is dit een goedkope manier om meer inzicht te krijgen.

Voor een nieuw catering systeem in het stadion kan een bestelpagina gemaakt worden voor de Flutter app. Omdat Flutter veel ingebouwde widgets heeft en op een hybrid manier werkt, kun je efficiënte development realiseren. Hiervoor heb je een backend nodig die alle bestellingen opslaat en doorstuurd naar een admin panel. Voor de communicatie kan Socket IO gebruikt worden. Voor de POC heb ik dit in een Node JS backend gedaan. Met als reden dat Socket IO origineel een JavaScript package is. Hierdoor is dit meteen compatible in NodeJS. Zo hoef je alleen in Flutter bepaalde dingen nog om te zetten.

Design

Tot op heden wordt er binnen de PSV app gebruik gemaakt van 2 soorten menu's die in elkaar zijn verwerkt, zo is er een sticky navbar onderaan de app en binnen deze navbar zit een hamburger menu die vervolgens alle andere features van de app laat zien. Uit de A/B test is gekomen dat gebruikers dit best verwarrend vinden dat er een menu in een menu zit, en ook dat er de tekst 'meer' bij het hamburger menu staat omdat het menu icoontje zelf al aantoont dat er meer te bekijken valt. In het design hieronder is de oplossing te zien voor deze verwarring, zo is het beter om de onderste navbar te gebruiken voor de belangrijkste features zoals 'tickets', 'loyalty' & 'seizoenkaart' en vervolgens een hamburger menu rechtsboven neer te zetten waar de fans op kunnen drukken om vervolgens meer features te bekijken.