

Adviesrapport crossplatform development



Flutter

Door: Sadek al Mousawi en Renas Khalil

Datum: 15 december 2024

Vak: Crossplatform Development

Inhoudsopgave

Inleiding	3
1. Shared Preferences	4
1.1 Reden van keuze:.....	4
1.2 Waarom Shared Preferences?	4
2. Google Maps en Location Packages	5
2.1 Onderzoek en huidige stand van zaken	5
2.2 Huidige bevindingen	5
2.3 Advies	5
3. Bronnenlijst	6

Inleiding

Dit rapport biedt een overzicht van de keuzes en overwegingen die zijn gemaakt tijdens de ontwikkeling van het project, met als doel een goed gefundeerd advies te geven over de gebruikte technologieën en de te nemen stappen in toekomstige sprints.

1. Shared Preferences

1.1 Reden van keuze:

Shared Preferences is door ons gekozen omdat dit in de lessen aan bod is gekomen en het een eenvoudige oplossing biedt voor het lokaal opslaan van gegevens in Flutter-applicaties. Binnen ons project wordt Shared Preferences specifiek gebruikt voor het opslaan van de **API-token & username** van de gebruiker. Deze token wordt gebruikt om:

- **Authenticatie:** Te controleren of een gebruiker is ingelogd.
- **Autorisatie:** Te bepalen of de gebruiker de juiste permissies heeft om acties zoals het aanmaken van events en teams uit te voeren.
- **Filteren:** Te filteren van een gebruiker om zijn teams & evenementen te tonen.

Door gebruik te maken van de API-token is het mogelijk om gebruikersinformatie veilig en efficiënt te valideren zonder dat er een aparte database lokaal hoeft te worden aangemaakt.

Kenmerk	Shared preferences	SQLite	Hive	Secure Storage
Simpliciteit	Hoog	Gemiddeld	Gemiddeld	Hoog
Prestaties	Hoog	Hoog	Hoog	Gemiddeld
Beveiliging	Laag	Gemiddeld	Gemiddeld	Hoog

1.2 Waarom Shared Preferences?

Gezien de eenvoud en het beperkte bereik van ons huidige project, voldeed Shared Preferences aan de eisen. Het stelt ons in staat om snel en eenvoudig de API-token lokaal op te slaan zonder complexiteit toe te voegen. Voor toekomstige uitbreidingen, waarbij meer veiligheid vereist is (bijvoorbeeld encryptie van tokens), kan Secure Storage worden overwogen.

Aanbevelingen voor toekomstig gebruik:

1. **Encryptie toevoegen:** Gebruik encryptiebibliotheken (bijvoorbeeld `flutter_secure_storage`) in combinatie met Shared Preferences om gevoelige gegevens zoals API-tokens te versleutelen.
2. **Data-uitwisseling:** Voor projecten waarbij data tussen verschillende apparaten gesynchroniseerd moet worden, kan een cloudgebaseerde oplossing zoals Firebase Realtime Database worden overwogen.
3. **Bewaking van opslaglimieten:** Houd rekening met het beperkte opslagformaat van Shared Preferences. Bij overschrijding kan data onvolledig of niet correct worden opgeslagen.
4. **Beveiligingsoverwegingen:** Hoewel Shared Preferences handig is voor eenvoudig lokaal gebruik, is het niet beveiligd tegen aanvallen zoals reverse engineering. Gebruik het alleen voor niet-kritieke gegevens.

2. Google Maps en Location Packages

2.1 Onderzoek en huidige stand van zaken

Tijdens de ontwikkeling van ons project hebben we onderzocht of de **Google Maps Flutter**-package en de **Location**-package geschikt zijn voor onze toepassing. Deze tools bieden de mogelijkheid om kaarten en locatie-informatie te integreren in onze applicatie. Hoewel deze packages krachtig en uitgebreid zijn, hebben we nog niet voldoende tijd gehad om hun volledige potentieel te beoordelen binnen onze specifieke context.

2.2 Huidige bevindingen

- **Google Maps Flutter:**
 - Sterk in het weergeven van kaarten en interactie met geodata.
 - Goede documentatie en brede ondersteuning.
 - Vereist API-sleutels en configuratie voor gebruik.
- **Location Package:**
 - Handig voor het verkrijgen van de huidige locatie van een gebruiker.
 - Ondersteunt zowel foreground- als backgroundlocatie-updates.
 - Vereist toestemming van de gebruiker en aanvullende beveiligingsmaatregelen.

2.3 Advies

We adviseren om in **Sprint 5** meer tijd te besteden aan het verdiepen in deze packages. Op basis van de eerste bevindingen lijken deze tools geschikt, maar een definitief advies kan pas worden gegeven na verdere implementatie en evaluatie. Mochten deze tools niet volledig aan de eisen voldoen, dan wordt geadviseerd om alternatieve oplossingen te onderzoeken, zoals:

- Mapbox (voor kaarten)
- Geolocator (voor locatie)

3. Bronnenlijst

flutter_secure_storage | *Flutter package*. (z.d.). Dart Packages.

https://pub.dev/packages/flutter_secure_storage

geolocator | *Flutter package*. (z.d.). Dart Packages.

<https://pub.dev/packages/geolocator>

google_maps_flutter | *Flutter package*. (z.d.). Dart Packages.

https://pub.dev/packages/google_maps_flutter

location | *Flutter package*. (z.d.). Dart Packages. <https://pub.dev/packages/location>

Saleh, M. (2023, 14 november). Local Storage Solutions for Flutter: Summary Guide -

Mahmoud Saleh - Medium. *Medium*. [https://medium.com/@mahmoudsaal7/local-](https://medium.com/@mahmoudsaal7/local-storage-solutions-for-flutter-summary-guide-83efd16c27bf)

[storage-solutions-for-flutter-summary-guide-83efd16c27bf](https://medium.com/@mahmoudsaal7/local-storage-solutions-for-flutter-summary-guide-83efd16c27bf)

Store key-value data on disk. (z.d.). Flutter.

<https://docs.flutter.dev/cookbook/persistence/key-value>