

Vorname	Nachname	Beruf	Dauer
Marten	Meißner	Fachinformatiker AE	07.22-07.24

Wochenbericht KW 43. (2023.10.23. - 29.)

Nach den letzten Anpassungen der Daten Logik in unserem „MeetingAlert“ Projekt, stießen wir immer wieder auf dasselbe Problem.

Es viel immer wieder auf die Termine korrekt erstellt wurden, aber mit unserer Erstellungslogik ließen sich die Probleme nicht lösen.

Probleme waren:

- Termine konnten immer nur sehr ineffizient erstellt werden.
- Sortierung der Termine, war sehr umständlich.
- Änderungen waren schwierig und wurden in der Logik immer wieder überschrieben.
- Die Baupläne für wiederkehrende Termine wurden, umständlich ausgelesen und mit viel

Branching unnötig komplex ausgewertet

- die ViewModel Objekte waren überladen mit doppelten Daten und unübersichtlich
- Theoretisch unendlich viele Termine erstellen und speichern, viele Ressourcen für

Suchen der VM's.

- Erstellen und suchen der richtigen Dates kostet viele Ressourcen
- Wenn wir neue Termine suchen oder berechnen, wollen wir Animationen anzeigen
- Animationen werden abgehackt, oder völlig übersprungen.

Nachdem wir uns im Team darüber beredet hatten, entschieden wir, dass wir an diesem Punkt nicht wirklich eine Verbesserung der Logik erhalten, wenn wir die Datenstruktur nicht optimieren.

Durch dieses Gespräch entschieden wir uns, einen Ladebildschirm zu initialisieren.

Denn unsere App startet immer mit den Standarddaten und anschließend, wenn unser erstes Event durchlaufen sind, dann erst die Einstellungen des Nutzers übernommen und wurden erst angezeigt, wenn der Nutzer mit der App interagiert hat und die GUI neu aufgebaut wurde.

Um diese Probleme zu lösen, entschieden wir uns einen Ladebildschirm zu bauen und wenn es nötig ist, die Animationen auf einen eigenen Dart Isolate (Thread) auszulagern.

Des Weiteren entschieden wir, nicht für jeden erstellten Termin ein Gaze Termin Objekt zu erstellen, sondern nur noch ein Zeitobjekt.

Da die Liste der Datenmodelle nicht so stark wächst wie die erstellten Terminzeit-Objekte, werden wir die Viemodels nun immer noch in einer Liste speichern, diese, enthält auch alle Informationen zu den Terminzeiten.

Zusätzlich bekommt unsere Datenlogik jetzt ein assoziatives Array, mit dem wir jetzt die Terminzeit-Objekte vorsortiert, abspeichern zu den jeweiligen Tagen an dem sie stattfinden.

Kontrolliert am: _____ Unterschrift : _____