

---

Vorname	Nachname	Beruf	Dauer
Marten	Meißner	Fachinformatiker AE	07.22-07.24

---

## Wochenbericht KW 23. (2023.05.29. - 06.04.)

---

Nachdem wir uns die letzten Wochen Gedanken gemacht haben, was das Konzept und die Funktionen der nächsten App sein wird und wie wir dieses Projekt umsetzen und in welchen Arbeitsschritten. Wurde diese Woche etwas praktischer, nachdem wir uns einig waren über die Funktionen und das Design, haben wir uns bei der Entwicklung der einzelnen Objekte, ganz nach dem Design Prinzip des MVC der Objektorientierten Programmierung gehalten. MVC steht für Models, Views, Controller und umschreibt die Abstraktion der einzelnen Bestandteile eines Programmes voneinander zu trennen, da dies die Wartbarkeit verbessert, den Austausch von einzelnen Objekten oder die Modellierung neuer Features vereinfacht. Zum Beispiel werden bei MVC ein Datenmodell erstellt, was für die Speicherung der Daten für ein bestimmtes Objekt dient. Dieses Objekt wird dann aber für das Frontend trotzdem neu erstellt, da so Boilerplate code verhindert wird und jedes Objekt oder Klasse atomar angepasst werden kann, ohne andere Teile des Programms zu blockieren oder die Funktionsfähigkeit eingeschränkt wird. In unserer App wäre dies der Fall bei der Datenklasse für einen Monat, diese Klasse enthält alle Funktionen einen Monat, Kalenderwoche oder den einzelnen Tag zu berechnen und dieses Objekt zu erzeugen und eventuelle Daten an den Appstate weiterzugeben. Auf der View Ebene haben wir eine Monthview Datei, in dieser wird die Grafische Oberfläche erzeugt und kann über einen Controller die Informationen aus der Datenklasse erhalten und dem Nutzer darstellen. Durch diese Abstraktion kann sichergestellt werden, wenn wir mal eine optische Änderung einplane, wird dies kein Einfluss auf die Datenklasse und die Logik im Backend haben, aber es kann auch eine Logik Anpassung stattfinden, ohne diese Änderung in jeder einzelnen Datei durchzuführen. Um diese Abstraktion aufrechtzuerhalten und diese einzelnen Objekte nicht aufeinander zuzugreifen und somit eventuelle Daten oder Informationsaustausch auszuschließen, wird ein Controller verwendet. Der Controller verwaltet die gesamte Logik und Interaktion mit den Objekten. Die Daten- und View-Klassen werden sehr schlank und simpel gehalten, um dort nicht viel Logik oder Interaktion zu erhalten, diese Eigenschaften werden dann in den Controller Objekten gesammelt und dort verwaltet, sodass wenn Änderungen passieren nur an einer Stelle angepasst werden muss, dass die einzelnen Objekte sich nicht gegenseitig aus Versehen manipulieren können und jede Interaktion im Programm über den Controller verwaltet wird. Weil wir uns an dem MVC Prinzip orientieren, nutzen wir in unserer App die Flutter Library BLoC, da diese Package diese Abstraktion von Data und View sehr gut erreicht, da jede Interaktion und State Anpassung über ein Bloc Event ausgelöst wird, im Event Funktionen und Interaktion nach fest vordefinierten Reihenfolge ausgeführt werden kann und am Ende des Events nicht ein State angepasst wird, sondern ein neuer State entsteht und basierend darauf die View und die Grafische Oberfläche neu aufgebaut und gerendert wird. Auf dieser Basis erstellten wir einen Appstate und banden das BLoC Package von Flutter ein und bauten unsere bisherige App so um, dass wir alle Änderungen und States über BLoC ausführen.

Kontrolliert am: \_\_\_\_\_ Unterschrift : \_\_\_\_\_