CoCo

Conditional Coloring

Problemstellung

Man kann sich den Vorlesungsplan bekannterweise in den Google-Kalender importieren lassen

Das Problem, welches vor allem dieses Semester mit den Wahlpflichtmodulen hinzugekommen ist, ist, dass viele Einträge hinzugefügt werden, an denen man gar nicht teilnimmt.

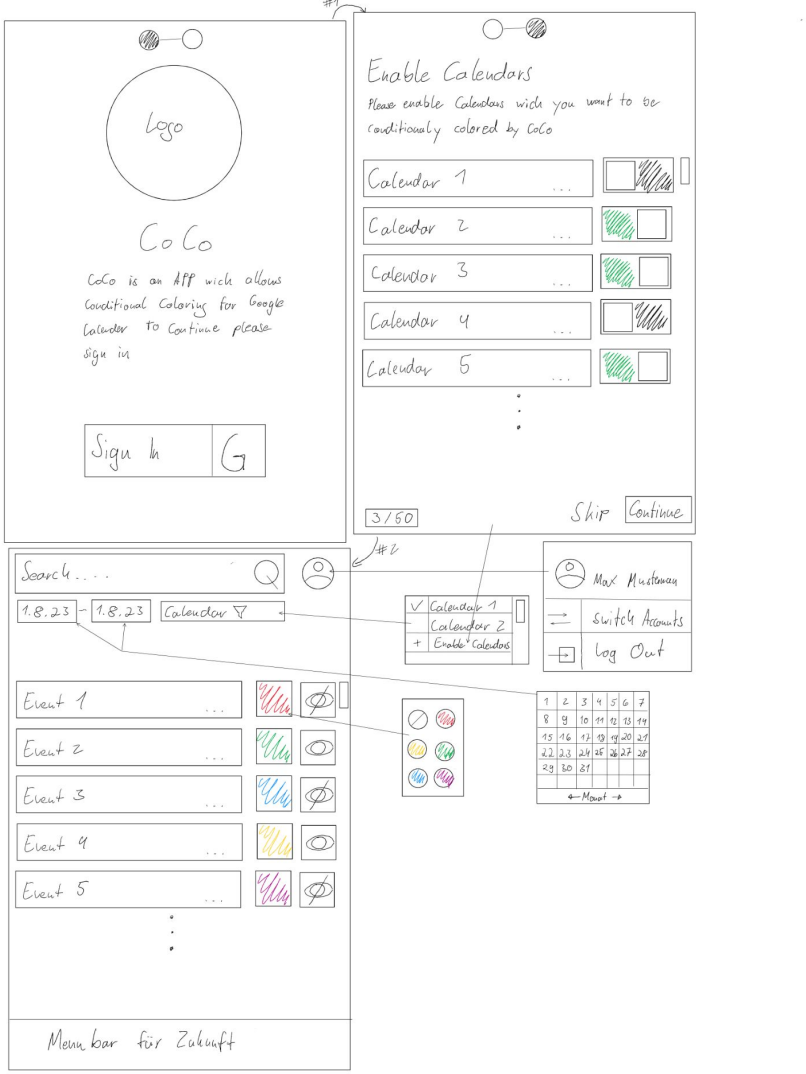
Außerdem besteht weiterhin das Problem, dass alle Vorlesungen mit der gleichen Farbe eingetragen werden.

Dieses Projekt versucht, diese Probleme zu beheben.

Zielstellung

Was muss die App für den Nutzer machen können?

- ändert Events im Google Kalender
  - mit Google acc verbinden
  - Berechtigungen, Authentifikation
- kann Kalender filtern
  - Abruf und Anzeigen der Kalender
  - Filteralgorithmus
- kann Events filtern
  - Abruf und Anzeigen der Events
  - Filteralgorithmus
- kann events ausblenden und einfärben
  - intuitive Interaktionen
  - farbwähler


Logo
CoCo
CoCo is an App wich allows conditional Coloring for Google Calender to Continue please sign in
Sign In
G
Enable Calendars
Please enable Calendars wich you want to be conditionally colored by CoCo
Calendar 1
Calendar 2
Calendar 3
Calendar 4
Calendar 5
3/50
Skip
Continue
Search...
1.8.23 - 1.8.23
Calendar
Calendar 1
Calendar 2
+ Enable Calendars
Max Mustermann
Switch Accounts
Log Out
Event 1
Event 2
Event 3
Event 4
Event 5
Menu bar für Zukunft

Pretotype

Storyboard

anmelden	konfigurieren	einfärben	ausblenden	filtern	
Google Account	Kalender	Rot	Events	nach Eventname	
	Event/Eigenschaft Paar	Gelb		nach Datum	MVP1
		Blau		nach Kalender	MVP2
		Grün			
		Türkis			

Nutzergedanken

	Szene 1, Öffnen der App	Szene 2, Anmelden	Szene 3, Kalender auswählen
Gedanke des Nutzers	Ich habe mir die App installiert und öffne Sie, ohne ein vollständiges Verständnis zu haben, was diese exakt tut.	Ich verstehe, was die App macht und möchte sie verwenden. Ich drücke also auf das einzigste Interaktionselement was mir Angeboten wird: der Google Login Button.	Ich sehe meine Google Kalender als Liste vor mir und kann mit einem Schalter diese zur Bearbeitung aktivieren oder deaktivieren, sowie fortfahren.
Aktion in der App	App öffnet den Willkommens-Screen, welche die App leicht verständlich erläutert und einen einzigen klaren Login-Button anbietet	Ein von Google standardisierter Login-Screen öffnet sich, und fragt alle notwendigen Berechtigungen ab und setzt den Nutzer in den Kalenderauswahl-Screen ab	Die jeweiligen Toggle-Switches werden grün für aktiviert oder grau für deaktiviert. Ein hervorgehobener “continue” button ist unten rechts.

Nutzergedanken

	Szene 4, Events	Szene 5, Filtern	
Gedanke des Nutzers	Nachdem ich meine Kalender ausgewählt habe und auf den “continue” button gedrückt habe, möchte ich meine Events bearbeiten.	Ich grenze das Datum mit den Datums-pickern ein, und suche mit der Searchbar nach gewissen Events.	
Aktion in der App	Der Event-Screen kommt auf, listet alle Events. Searchbar und Datums-picker am oberend Rand. Ein/Ausblend-button und Color-picker an den jeweiligen Events.	Die Liste der Events passt sich dynamisch an die Eingrenzungen an.	

Analysis Paralysis vermeiden

=

Anfangen

Architektur und Struktur

Problem:

Live Updates und per Nutzer Speicher welcher immer abrufbar sein muss

Eigener Server?
Background Aktivität?
viel in zu Kurzer Zeit?

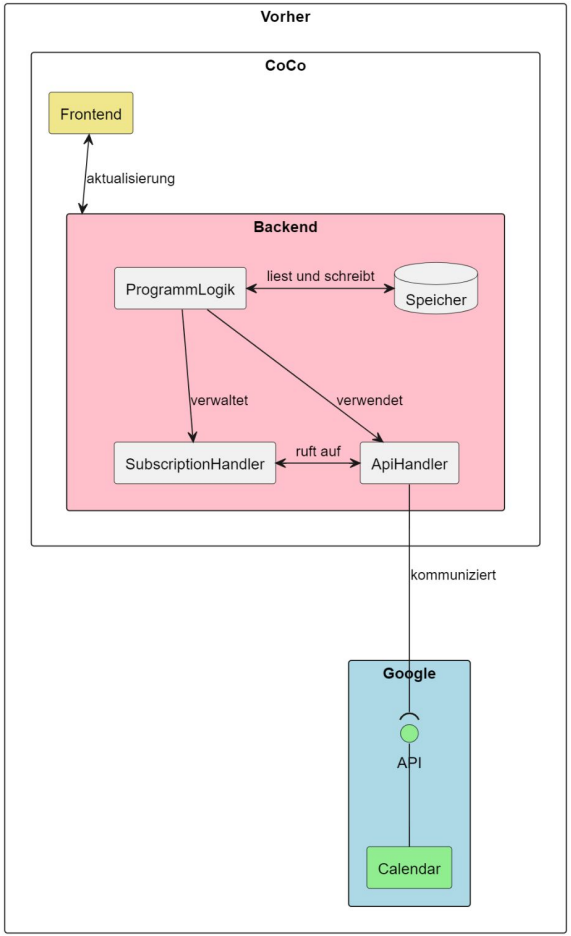
Lösung:

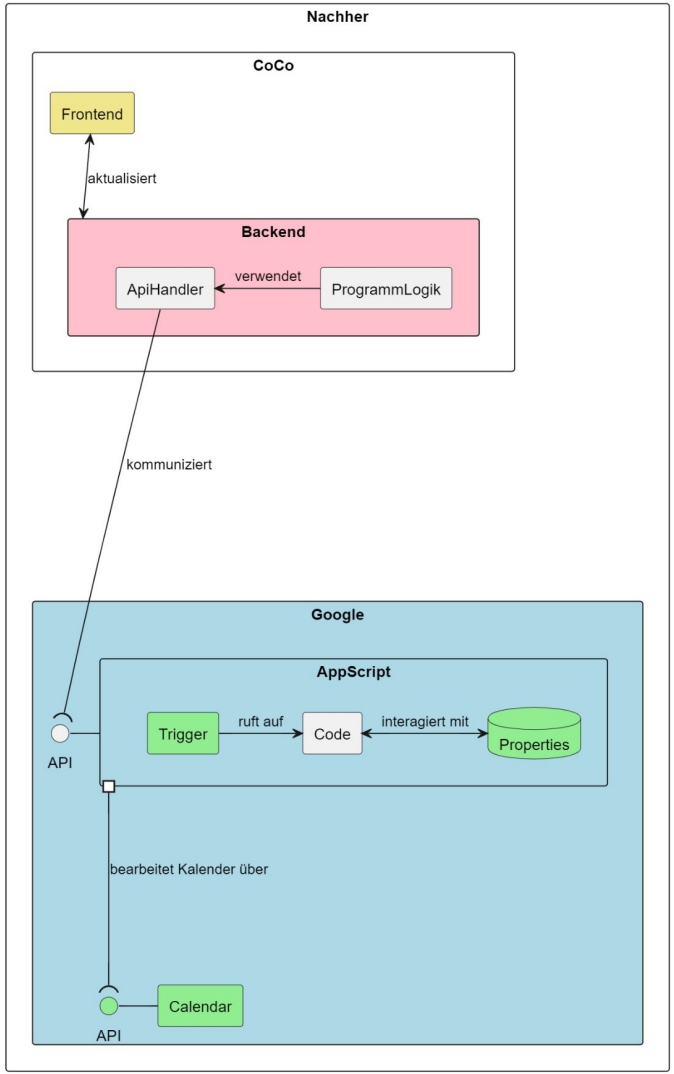
FAIL EARLY!

Schon im Designprozess Überladung festellen um nach alternativen zu suchen

In der Praxis: AppScript

Archite


Vorher
CoCo
Frontend
aktualisierung
Backend
ProgrammLogik
liest und schreibt
Speicher
verwaltet
verwendet
SubscriptionHandler
ruft auf
ApiHandler
kommuniziert
Google
API
Calendar


Nachher
CoCo
Frontend
aktualisiert
Backend
ApiHandler
verwendet
ProgrammLogik
kommuniziert
Google
AppScript
Trigger
ruft auf
Code
interagiert mit
Properties
API
bearbeitet Kalender über
Calendar
API

AppScript als API für CoCo

Was wurde umgesetzt?

- Google Calendar API abstrahieren
- Development + Testing in Postman
- Als separates Repository
- JSON
- Endpunkte:
  - Kalender Registrierung
  - Nutzer Property Verwaltung
  - Events
  - Kalender

-> Wie Integration in App?

Wichtigste Aspekte:

- Simpel
  - “App in Mind” Endpunkte
  - Kalender-Aktionen ausführen
- REST orientiert
  - HTTP
  - JSON
  - URIs

Problem:

CORS

Lösung:

Hinzufügen eines zusätzlichen Run-Arguments (--web-browser-flag "--disable-web-security")

Front End

Schritt 1: Pretotype in Mockup App umsetzen

- 3 Screens
- keine Funktionalität
- aber Interaktivität
- so nah wie möglich zum Zielaussehen
- Beispieldaten einbinden
- Feedbackloop mit Team

Schritt 2: Funktionalität einbinden

- erst Möglich, wenn Funktionalität vorhanden
- vorraussichtlich letzter Schritt des Projektes

(mir) Wichtige Aspekte:

- gutaussehend
  - Font
  - Theme
- Intuitiv
  - visuell lenkende Elemente
  - bekannte Elemente
- keine Menü- / Einstellungspage
  - ist hässlich
  - verkompliziert viel
  - unterbricht “Workflow”

Front End

Schritt 1: Pretotype in Mockup App umsetzen

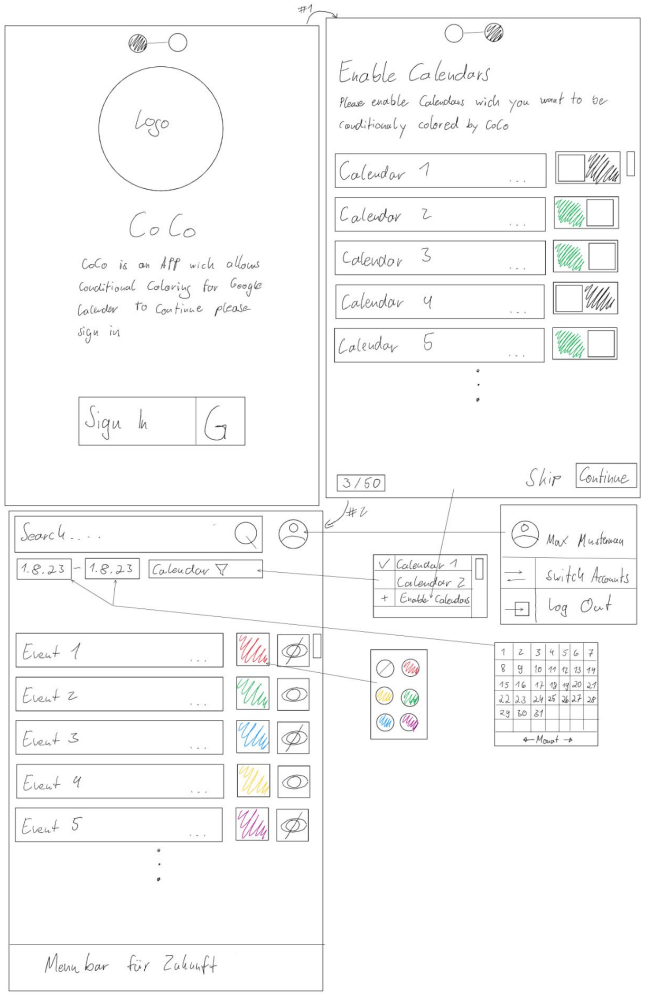
- 3 Screens
- keine Funktionalität
- aber Interaktivität
- so nah wie möglich zum Zielaussehen
- Beispieldaten einbinden
- Feedbackloop mit Team

Schritt 2: Funktionalität einbinden

- erst Möglich, wenn Funktionalität vorhanden
- vorraussichtlich letzter Schritt des Projektes

(mir) Wichtige Aspekte:

- gutaussehend
  - Font
  - Theme
- Intuitiv
  - visuell lenkende Elemente
  - bekannte Elemente
- keine Menü- / Einstellungspage
  - ist hässlich
  - verkompliziert viel
  - unterbricht “Workflow”


#1
Logo
CoCo
CoCo is an App wich allows conditional Coloring for Google Calendar to Continue please Sign in
Sign In
G
Enable Calendars
Please enable Calendars wich you want to be conditionaly colored by CoCo
Calendar 1
Calendar 2
Calendar 3
Calendar 4
Calendar 5
3/50
Skip
Continue
#2
Search...
Max Mustermann
Switch Accounts
Log Out
1.8.23 - 1.8.23
Calendar
Calendar 1
Calendar 2
+ Enable Calendars
Event 1
Event 2
Event 3
Event 4
Event 5
Month
Menu bar für Zukunft

Pretotype

Mock-Up App

Live Demo

Problem:

Unklarheit, warum welches Element eingebaut werden sollte

Lösung:

“Nutzergedanken” in Textform niedergeschrieben

Problem:

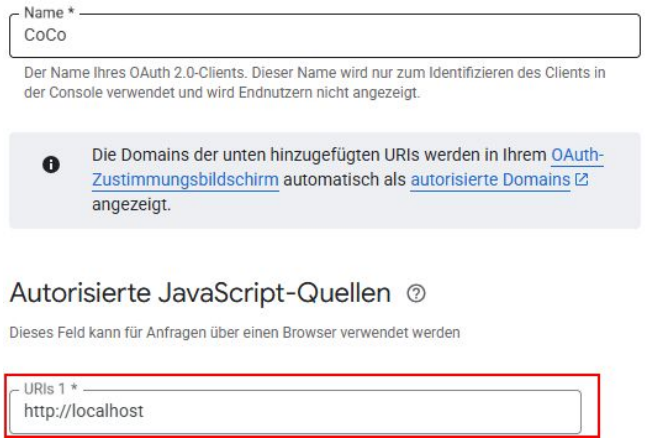
Kleines Projekt &
Jeder arbeitet an jeder Datei
-> viele merge Konflikte

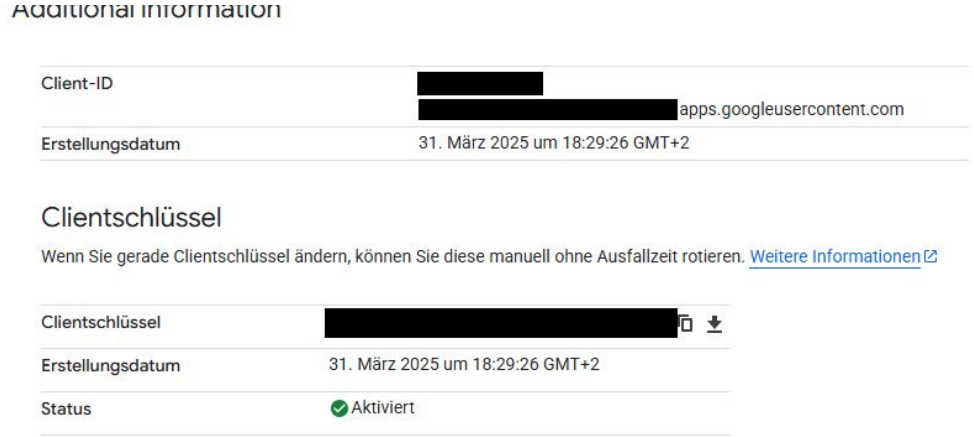
Lösung:

Klare Absprache, wer was wann bearbeitet

Authentifikation

Google Authentication (Web)

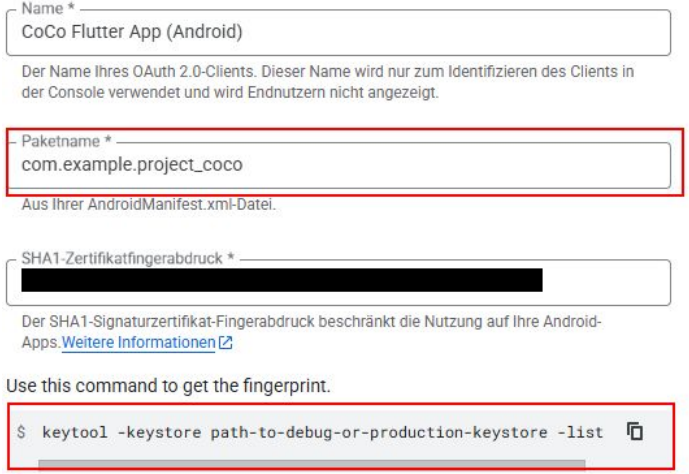




- Autorisierte JavaScript-Quellen einrichten (Für Entwicklung http://localhost)
- Hinzufügen eines Meta-Tags in der “web/index.html”:

```
<meta name="google-signin-client_id" content="Client-ID.apps.googleusercontent.com">
```

Google Authentication (Android)



- Generieren eines SSH-Fingerprints durch das keytool (vorinstalliert mit Android Studio)
- Hinzufügen der Client-ID JSON in “android/app”

LIVE DEMO

CoCo

Fragen?