Kirmes App

Alexander Brauneck

Inhalt

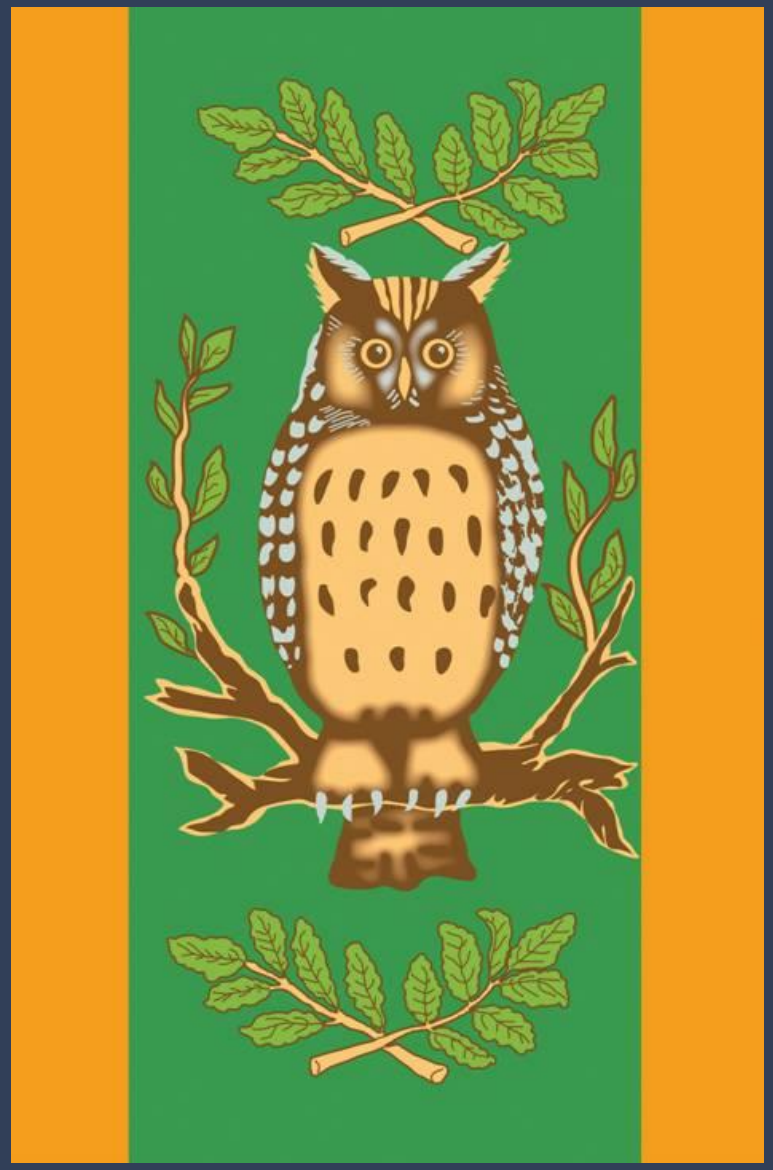
- Auftrag, Idee
- Aufbau der App:
  - KirmesView
  - KirmesViewModel
  - KirmesModel
  - Backend
  - URLSession – GET
  - PopupView
  - CurrencyTextField
  - URLSession – POST
  - Device Rotation
- Ausblick
- Links

Auftrag, Idee

- Schatzmeister von Kirmesgesellschaft
  - Sehr einfache App, mit der Bons verkauft werden
  - Summe errechnen, Rückgeld berechnen
  - Umsätze einsehbar
- Vorgänger auf Android von 2019 konnte nur Summe berechnen

-> Backend mit Preisliste, Umsätzen

-> Frontend einfach strukturiert, einfache Bedienung



Aufbau der App: KirmesView

- BottomBarView
  - Summe
  - Zahlen Button
- ItemView
  - PlusMinusButton
  - Text, Preis, Anzahl

Aufbau der App: KirmesViewModel

- Hat das Model
- Hat die URLs
- Hat die Funktionen für Plus/Minus Button und zahlen
- Lädt die KirmesItems aus der URL

Aufbau der App: KirmesModel

- Hat die Logik für die Funktionen vom ViewModel
- Hat die Liste der KirmesItems
- Berechnet die Summe
- Verändert die aus der URL geladenen Items, dass sie für das Frontend verwendbar sind.
  - Beispiel: Im Backend ist ein Hex-Code für die Farbe des Items hinterlegt, dieser wird hier zu einem UIColor umgewandelt.

Aufbau der App: Backend

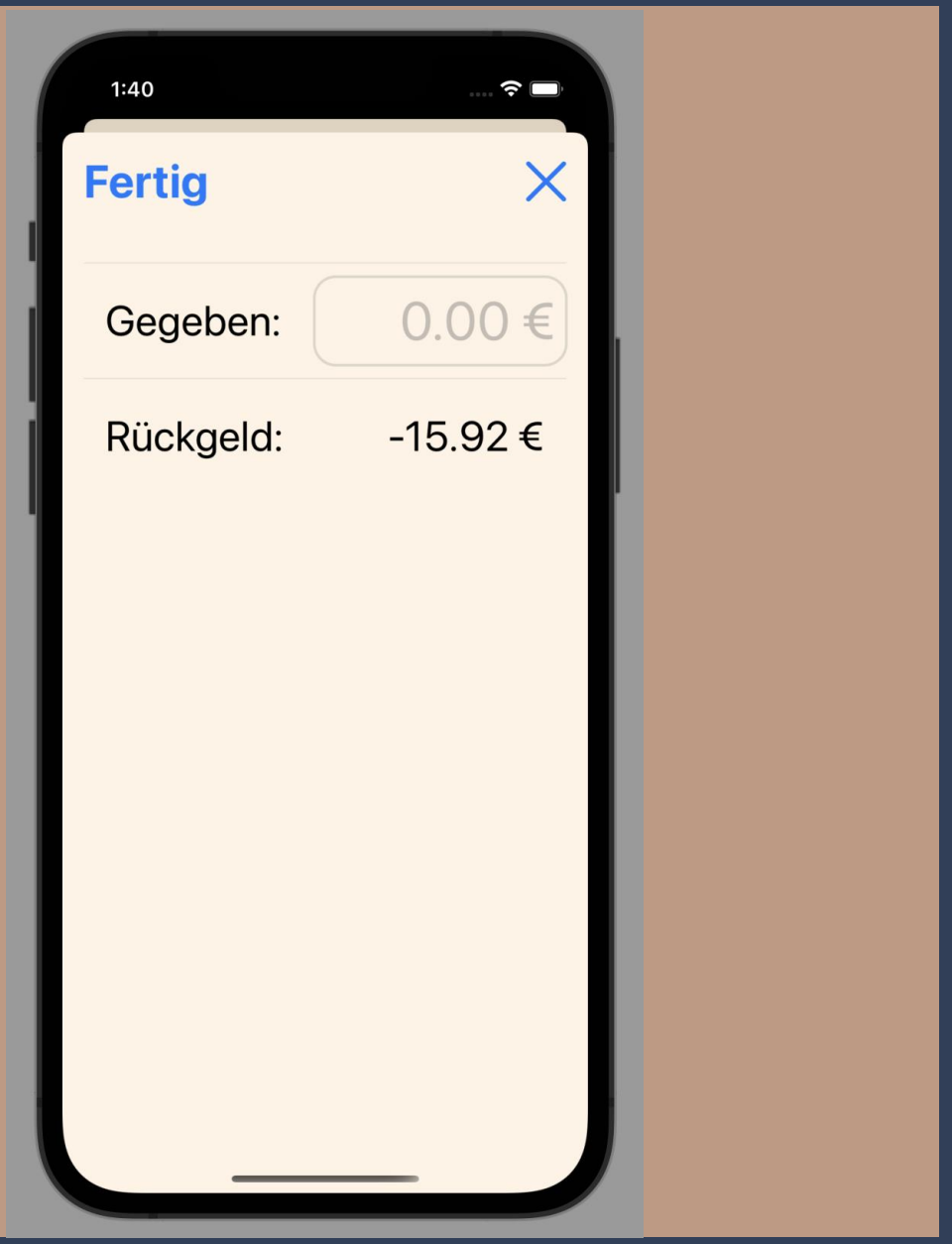
- Node.js, Typescript
- Simple REST Schnittstellen
  - Für KirmesItems
  - Für Quittung/Umsätze
- Sebstgehostet auf IP-Adresse von Windows PC
- Für KirmesItems: http://localhost/kirmes/items/
- Für Quittung: http://localhost/kirmes/quittung/

Aufbau der App: URLSession - GET

- Große Probleme die URL auszulesen im Model
- Viele Videos, alle anders, kein Ergebnis
- Der Typ musste so angepasst werden, dass alles aus der Item – URL gelesen werden konnte mit JSONDecoder, dann konnte ich darin nach dem suchen, was die App braucht.
- Das Ergebnis des JSONDecoder:
  -> [BackendKirmesItemsEnum]
  -> [[BackendKirmesItem], BackendKirmesItems]
  -> [[BackendKirmesItem], BackendKirmesItems.kirmesItems]

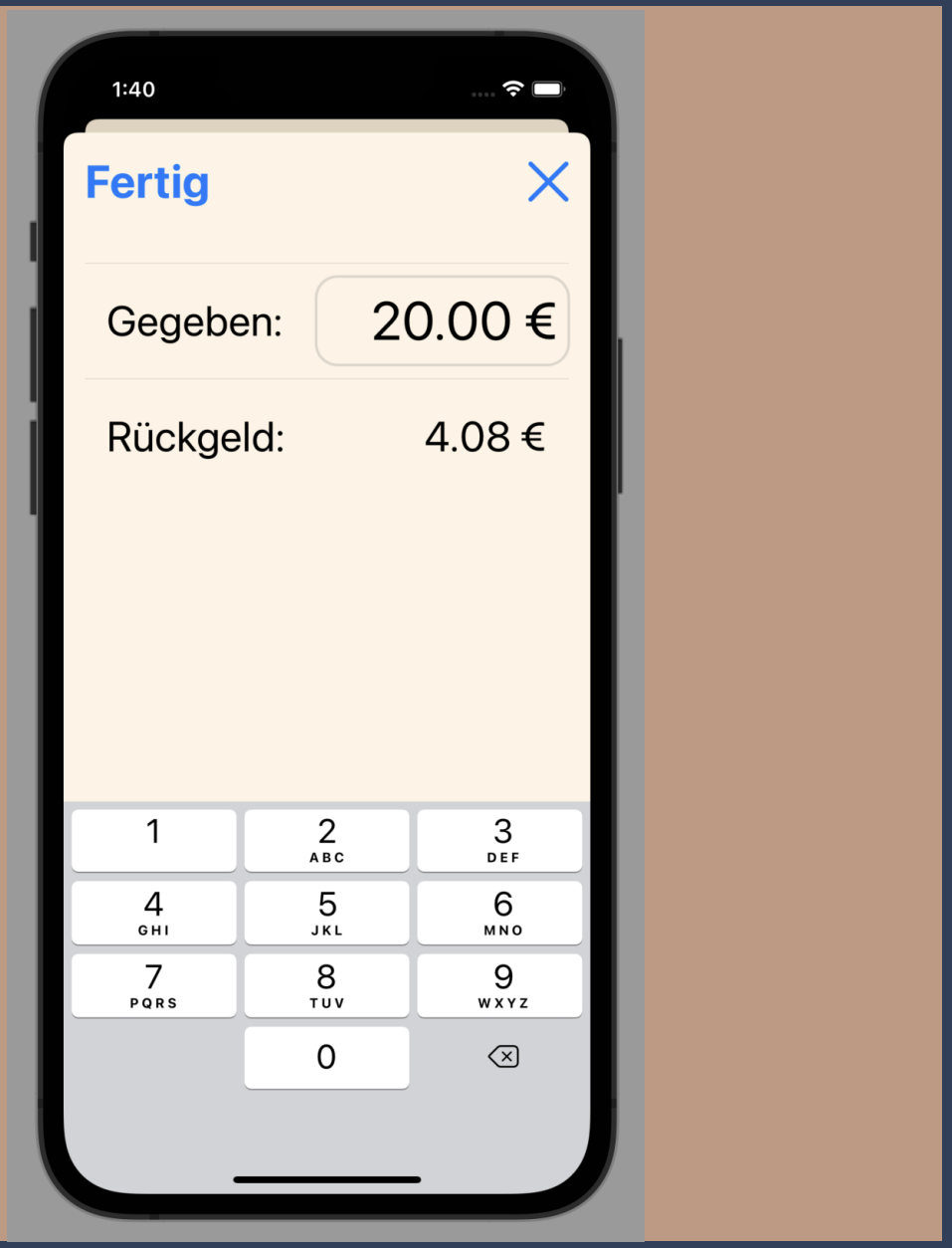
Aufbau der App: PopupView

- Wird angezeigt wenn eine @State var von der KirmesView = true ist.
- “Zahlen“ Button setzt var auf true
- Aus der Summe und dem „Gegeben“-Wert wird das Rückgeld berechnet.
  - CurrencyTextField für Gegeben-Wert
- X schließt das Popup und durch „Fertig“ wird ein Umsatz ans Backend geschickt.



Aufbau der App: CurrencyTextField

- Das Textfeld hat einen NumberFormatter
  - .currency
  - Währungszeichen „€“
  - Nachkommastellen: 2
- Öffnet Systemtastatur nur mit Zahlen (.numpad)
- Eingabe von hinten nach vorne
- Updatet nach jeder Eingabe
- Setzt Komma automatisch



Aufbau der App: URLSession - POST

- Aus der ItemList wird der Name und die Anzahl der Items ausgelesen deren Anzahl > 0 ist
- Diese werden in einen String geschrieben
- Und der wird an das Backend geschickt und in ein .json File hinterlegt

Aufbau der App: Device Rotation

- Problem: Im Landscape-Modus überdeckt die Tastatur einen Teil der View
- Lösung: View ändert sich, wenn Landscape die aktuelle Device Rotation ist
- Aus einem VStack wird ein HStack

Ausblick

- Verwendung bei Kirmes in 2 Wochen
- Individuelle Änderungen: Größen, Farben, Schrift, etc.
- Optimierung für bestimmtes Gerät
- Backend hosting
- WebUI für Backend
- Nach Android migrieren
- Wünsche Schatzmeister

Links

- Help:
  - CurrencyTextField: https://benoitpasquier.com/currency-textfield-in-swiftui/
  - URLSession – GET: https://app.quicktype.io
  - Google, YouTube

- Github Link:
  - https://github.com/AlexanderBrauneck/KirmesApp