SaveUp-App

Von Ege, Lenny, Beren, und David

Inhaltsangabe

- Ziel und Ausgangslage
- Design und Mockups
- App-Entwicklung und Funktionen
- MVVM-Struktur
- Code-Qualität und Fehlerbehandlung
- Testing und Dokumentation
- Live-Demo
- Fazit und Reflexion

Ziel und Ausgangslage

Ziel der App

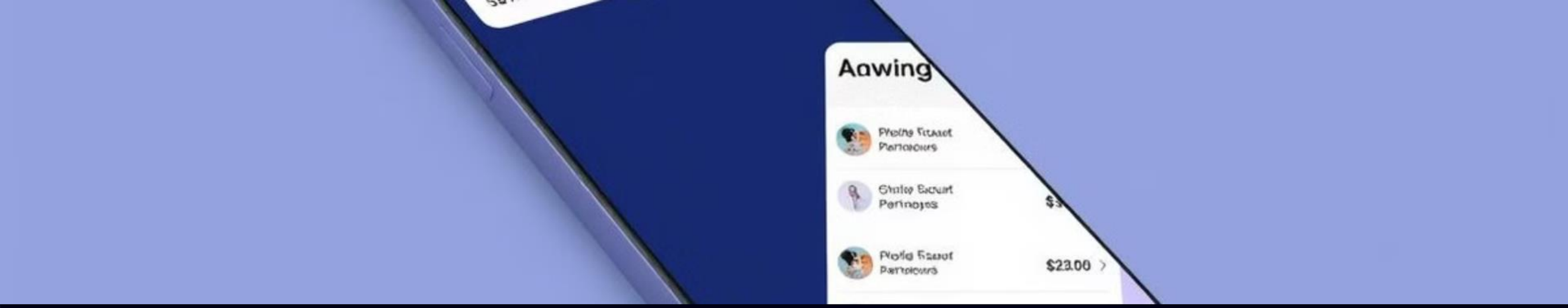
Entwicklung einer App, die hilft, durch Verzicht auf kleine Ausgaben zu sparen.

Zeigt den gespeicherten Betrag aus diesen Verzichtsprodukten an.

Ausgangslage

Die App hilft Benutzern, ihre kleinen Ausgaben zu überwachen und motiviert sie, Geld zu sparen.

Ermöglicht die Eingabe von Artikeln, die durch Verzicht gespart wurden, und den Preis dieser Artikel.



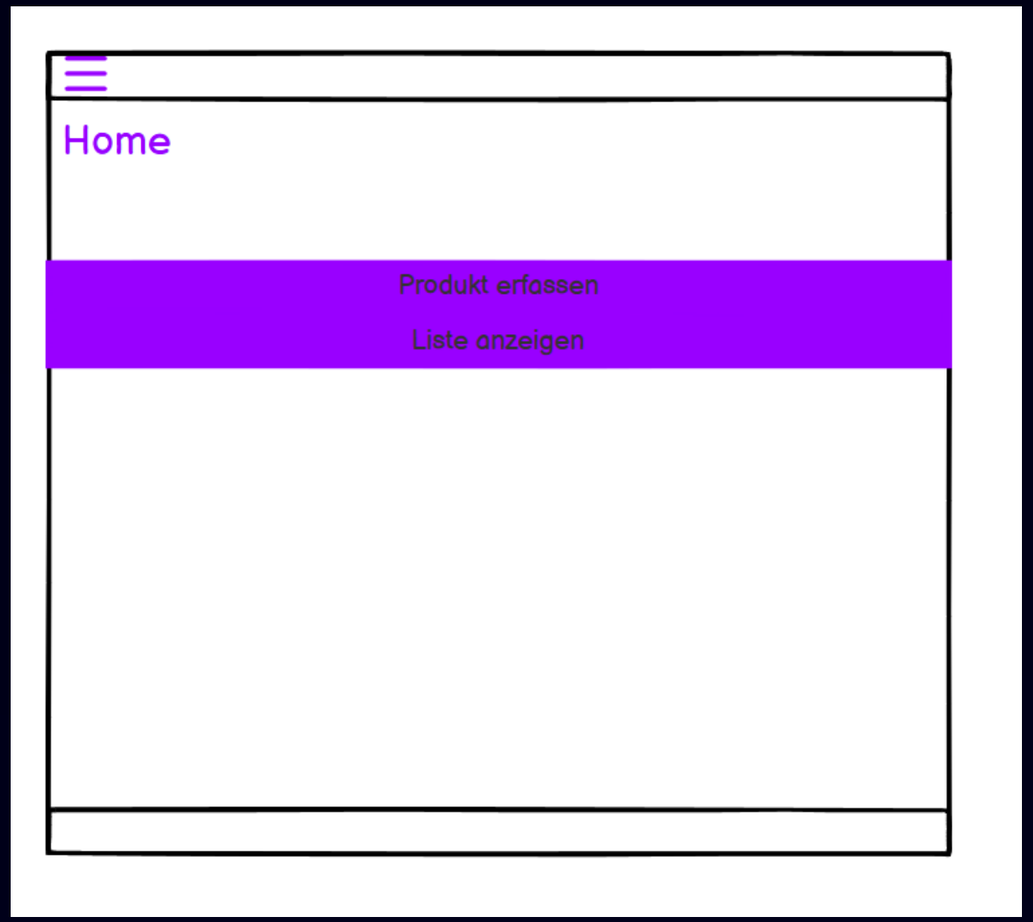
Design und Mockups

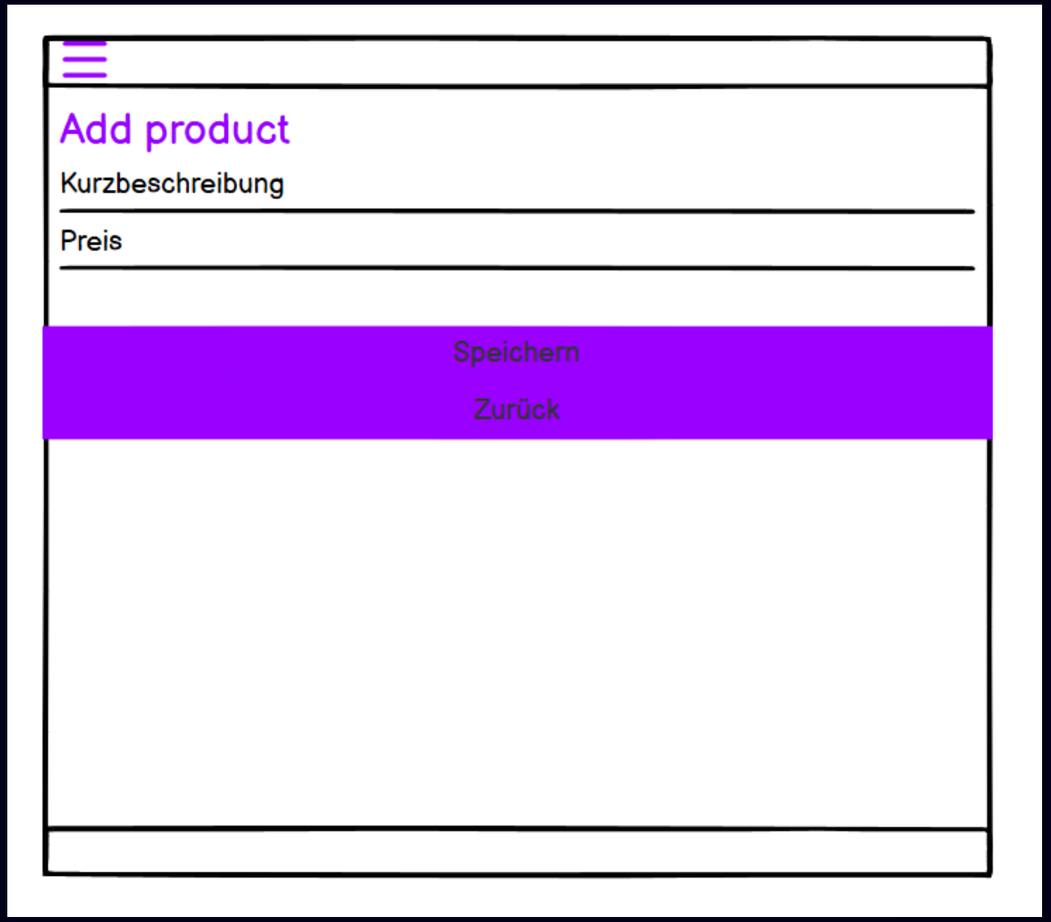
GUI-Design

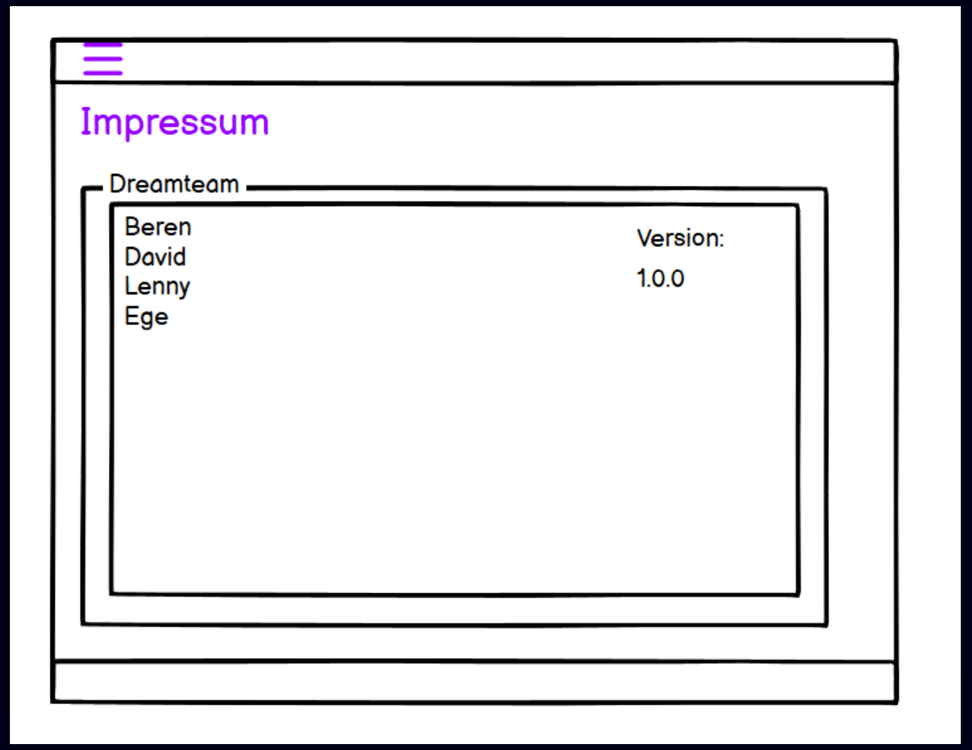
App-Icon

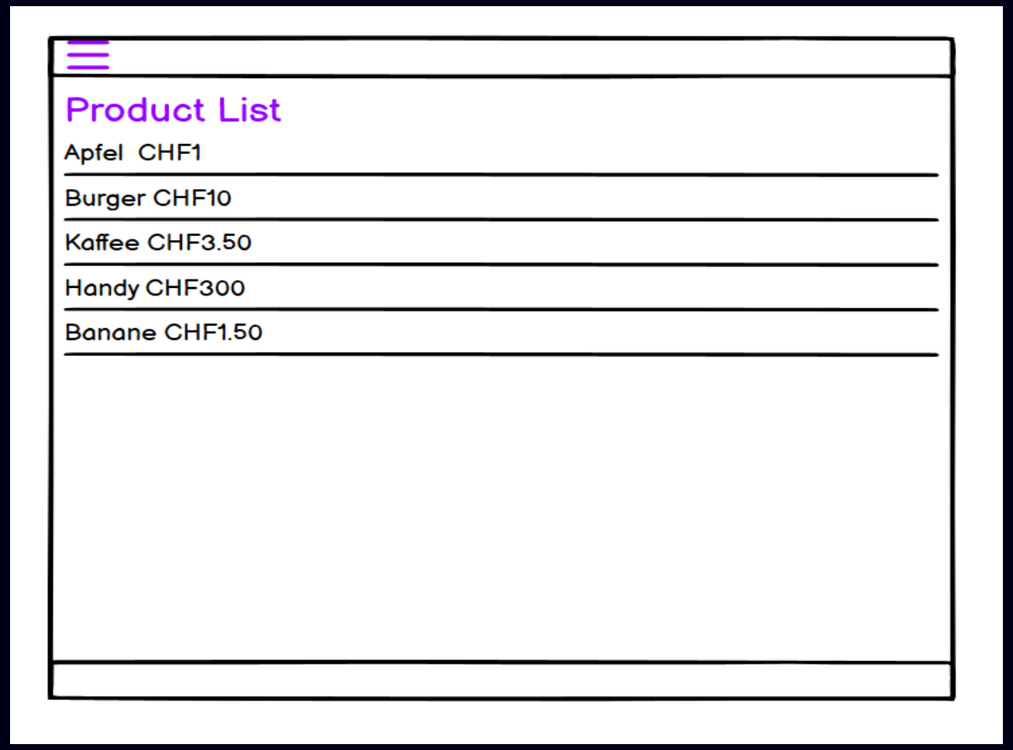
Ergonomie

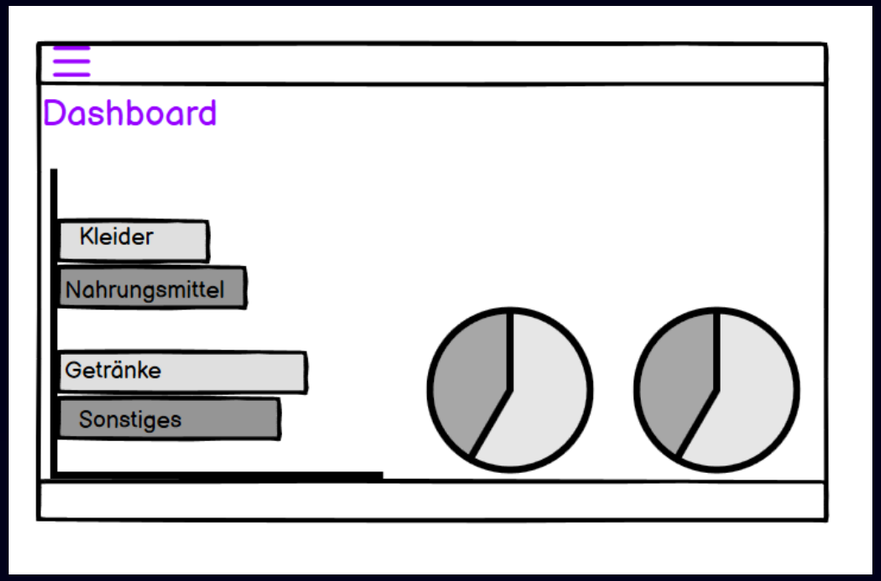
Intuitive Benutzerführung durch einfache Navigation und große Schaltflächen.

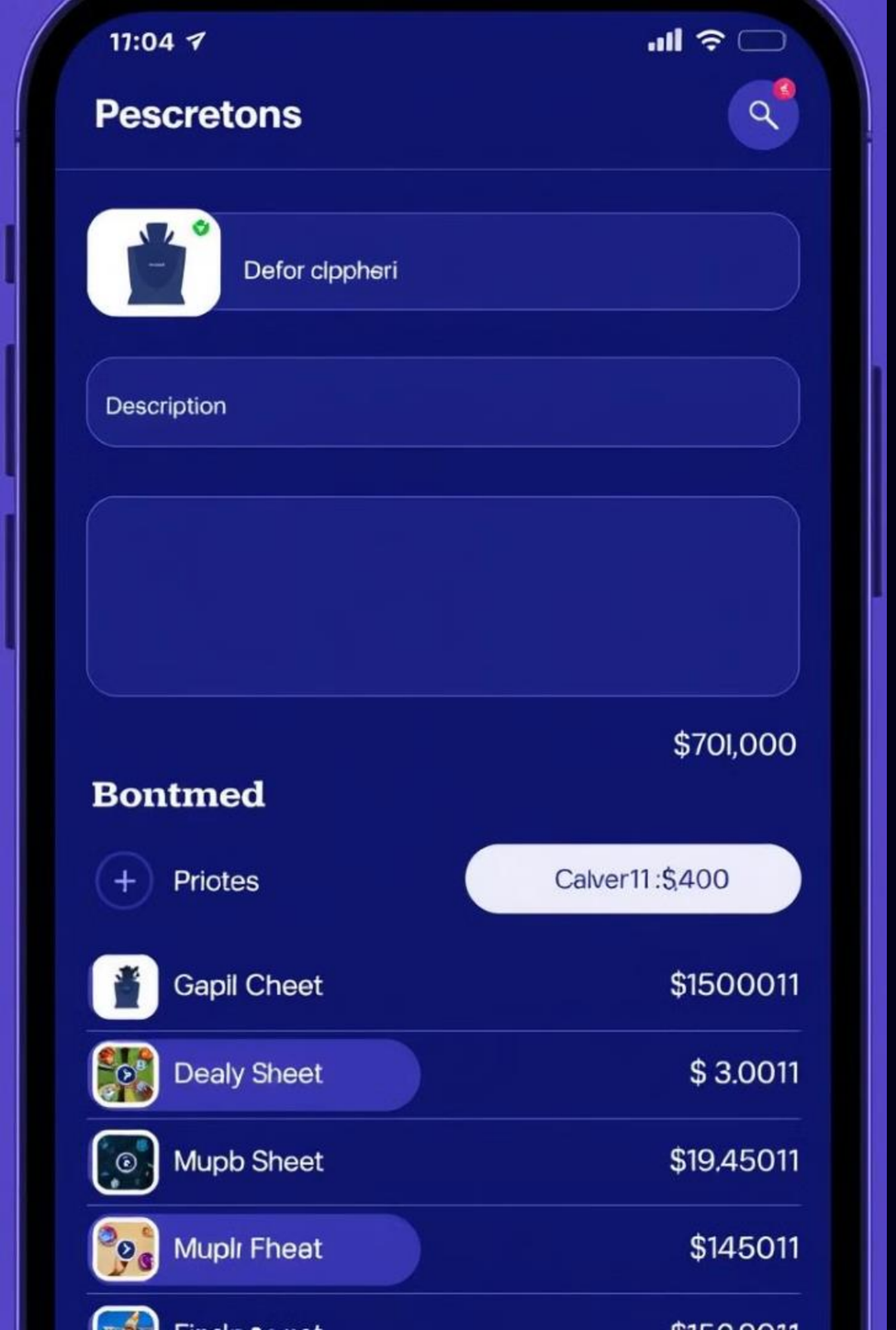

Home
Produkt erfassen
Liste anzeigen


Add product
Kurzbeschreibung
Preis
Speichern
Zurück


Impressum
Dreamteam
Beren
David
Lenny
Ege
Version:
1.0.0


Product List
Apfel CHF1
Burger CHF10
Kaffee CHF3.50
Handy CHF300
Banane CHF1.50


Dashboard
Kleider
Nahrungsmittel
Getränke
Sonstiges



App-Entwicklung und Funktionen

1 Produktbeschreibung und Preis erfassen

Mindestens 2 Eingabefelder für Produktbeschreibung und Preis.

2 Gesamtbetrag anzeigen

Summe der Preise der gesparten Produkte.

3 Speicherung und Abruf der Daten

Im ViewModel.

MVVM-Struktur

Model

Speichert die Produktinformationen und die Berechnungen (z. B. Produktname, Preis).

ViewModel

Verwaltet die Logik, wie das Hinzufügen von Produkten und das Berechnen des Gesamtbetrags.

View

Das XAML-Layout, welches die Benutzeroberfläche darstellt und die Bindungen zu den ViewModel-Eigenschaften enthält.

Code-Qualität und Fehlerbehandlung

Codequalität

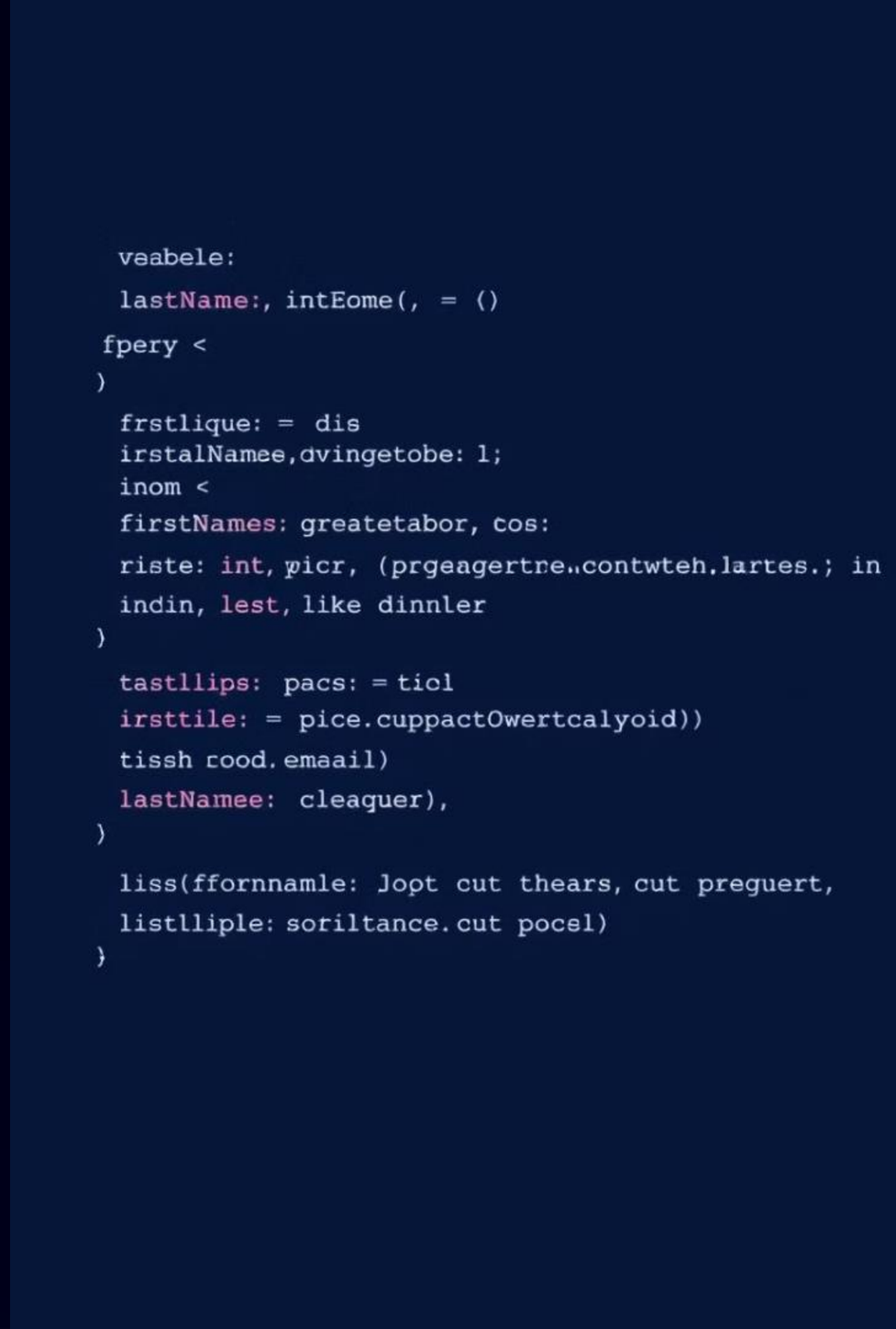
Saubere Struktur: Trennung von Logik und UI (MVVM).

Leserlicher Code: Kommentare und klare Benennung der Variablen und Methoden.

Fehlerbehandlung

Validierung: Überprüfen, ob die Eingabewerte (Preis und Beschreibung) korrekt sind.

Fehlermeldungen: Informative Fehlernachrichten, wenn die Eingaben ungültig sind.



2	Test Colore	Jiscrmtablee to Smiour	Insprat Scatent/ Cabur	Memstagcican to folgni Memdsiny	Moronstdbut are nor Cant faolgur Hettey'
4	Test Colsse	Jiscrmtablee at Smdonr	Insprat Scatent/ Cabur	Memstagciem of folourr Desndaing'	Woronstdbut are nor test Foolgur Mettey'
5	Test Colsse	Uscrmtablee to Smiour	Accnest Scatent/ Cutpur Heendaing	IAapestagcican of folgur Neendoing	Weronstdbut are nor test Foolgur Hettay'
5	Test Calere	Jiscrmiablee to Smionr	Insprat Scatent/ Cabur	IAsrmstegcican f Colbur Merklainy	Moronstdbut are rest tent Foolgur Mettey'
6	Ubeat Rejer	Jistrmiablee to Smiour	Inspert Sratent/ Cabur	Vierrnstagciem of folbur Meendoing	Weronstdbut are nor eert foolgur Nettey'
		Wacdalles:	Insport teplacation	Mpert ir orlyilee:	Acoigat dble inorhayilee:

Testing und Dokumentation

1

Manuelle Tests

Überprüfen der Funktionalität der Eingabefelder und der Listendarstellung.

2

Testplan

Dokumentation der Tests, um sicherzustellen, dass alle Funktionen korrekt arbeiten.

3

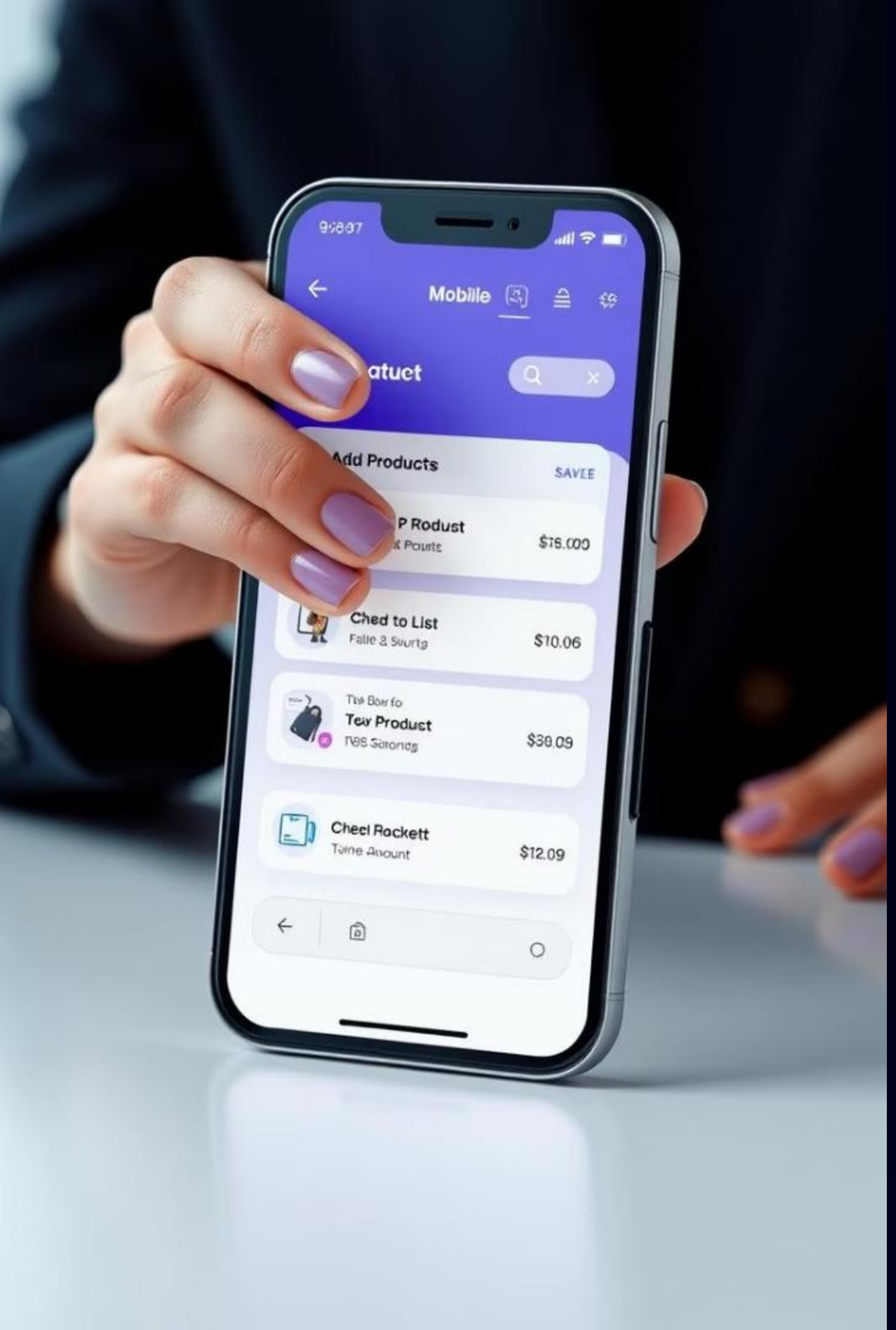
Dokumentation

Code-Kommentare und Projekt-Dokumentation mit Zielen und Arbeitspaketen.

4

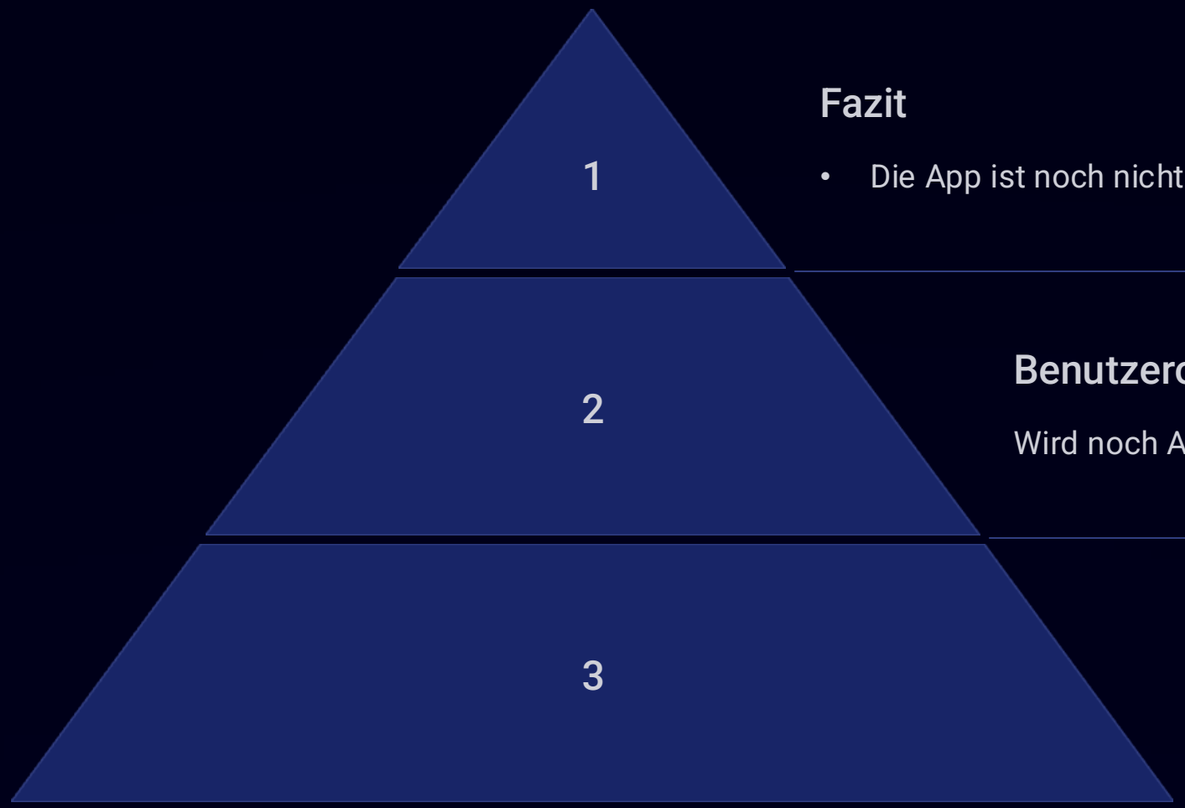
IPERKA

Verwendung von IPERKA zur Planung der Schritte und Aufgaben.



Live-Demo

Fazit und Reflexion



Fazit

- Die App ist noch nicht vollständig

Benutzeroberfläche

Wird noch Angepasst das es Benutzerfreundlich ist

Lessons Learned

Herausforderungen und Lösungen beim Implementieren der MVVM-Struktur.

Erkenntnisse aus dem Testen und der Optimierung der Benutzeroberfläche.


The End