Modul322_Omda_Maric – Projektdokumentation (IPERKA)

---

1. Informieren (Ausgangslage & Ziele)

Ausgangslage:
In der Lehre im Modul 322 soll eine plattformübergreifende App entstehen, die auf Windows, macOS, iOS und Android läuft. Bisher existieren unterschiedliche native Prototypen, die gewartet werden müssen. Ein einheitliches MAUI-Projekt soll diese Fragmentierung aufheben.

Ziele:

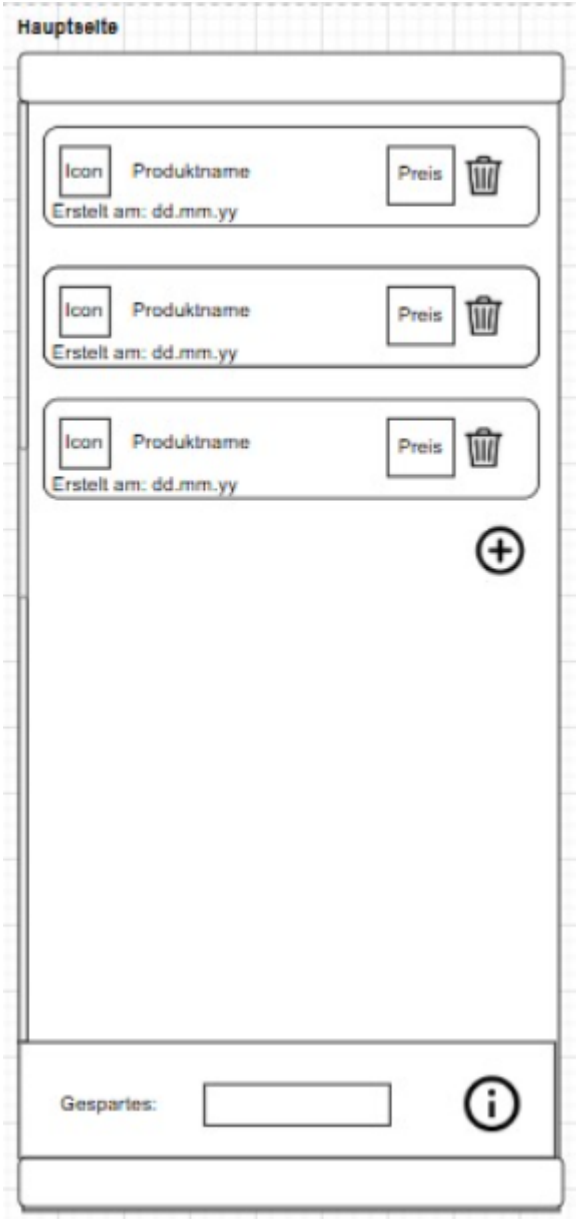
- **Primärziel:** Entwicklung einer .NET MAUI App mit einheitlicher Codebasis für alle vier Plattformen.
- **Nebenziele:**
  - MVVM-Architektur einführen
  - Wiederverwendbare UI-Komponenten gestalten
  - Automatisierte Build- und Testabläufe sicherstellen

---

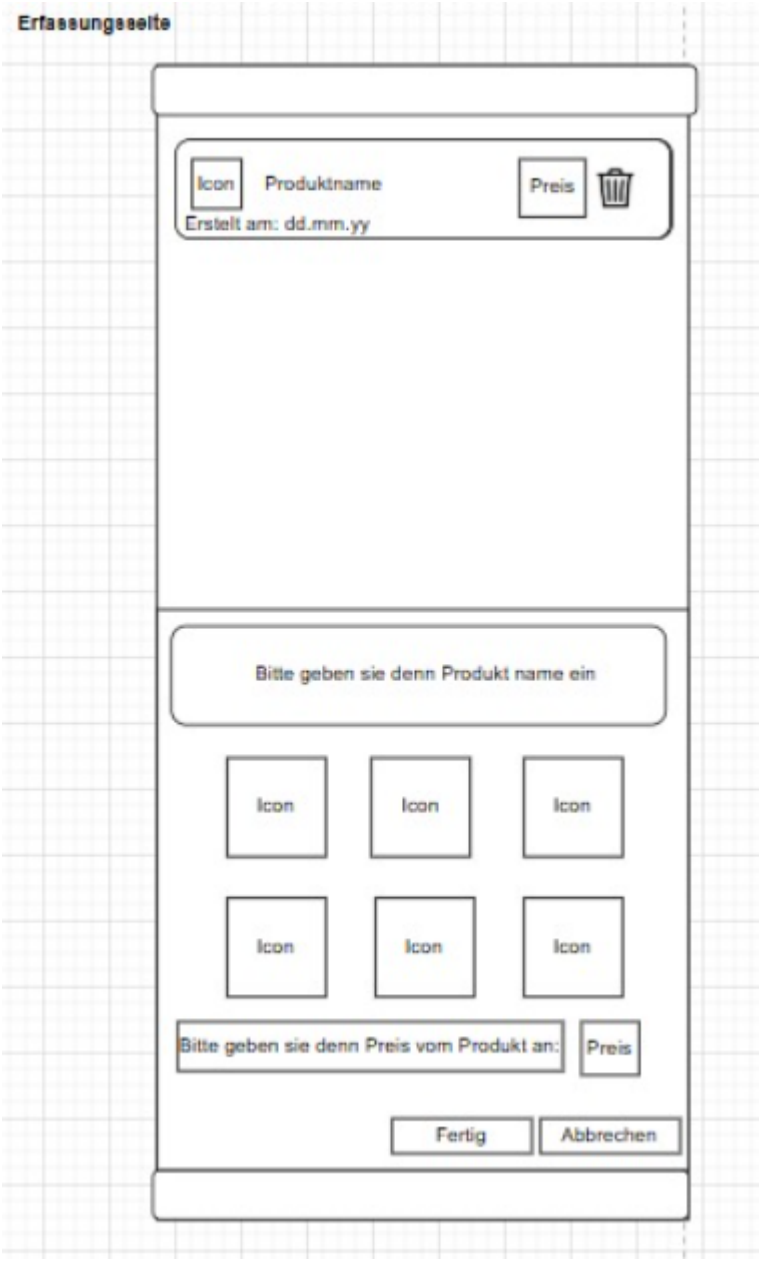
2. Planen (Mockups / Wireframes)

Hier siehst du die entworfenen Screens als Wireframes:

1. **Hauptseite**
   - Liste aller Produkte
   - Anzeige von Name, Erstellungsdatum, Preis
   - Button zum Anlegen neuer Produkte



1. **Erfassungsseite**
    - Formular zur Eingabe von Name, Icon und Preis
    - Validierungs-Hinweise
    - Buttons „Fertig“ und „Abbrechen“



2. **About-Seite**
    - Informationen zu Design und Programmierung
    - Rechtliches (Impressum, Datenschutz)

3. Entscheiden (Arbeitspaketplanung & Gantt-Chart)

Die Arbeitspakete wurden so zusammengefasst und terminiert, dass alle Aufgaben innerhalb von 7 Tagen abgeschlossen werden können:

Arbeitspaket	Dauer	Verantwortlich	Tag
Analyse & Anforderungsdefinition	1 Tag	Omda	Tag 1
UI-Entwurf & Wireframes	1 Tag	Maric	Tag 2
Implementierung Models & Services	2 Tage	Omda	Tag 3–4
UI-Implementierung (Views & ViewModels)	2 Tage	Maric	Tag 5–6
Testing, Bugfixing & Abschluss	1 Tag	Omda & Maric	Tag 7

4. Realisieren & Kontrollieren (Testplan & Protokoll)

4.1 Testplan

Testfall Nr.	Funktion	Erwartetes Ergebnis	Testumgebung
1	App-Start	App startet ohne Fehler, Hauptseite wird angezeigt	Windows 10, Visual Studio Debug
2	Neuen Eintrag anlegen	Eintrag erscheint in Liste, Felder korrekt gespeichert	Android Emulator (API 31)
3	Eingabevalidierung Name	Fehlermeldung bei leerem Namen	iOS Simulator (iOS 16)
4	Löschfunktion	Eintrag wird aus der Liste entfernt	macOS Monterey
5	Plattformwechsel	UI passt sich der Plattform an	Windows / Android / iOS

4.2 Testprotokoll

Datum	Tester	Testfall Nr.	Ergebnis	Bemerkung
22.05.2025	Omda	1	bestanden	Schnellere Ladezeit beobachtet
23.05.2025	Maric	3	fehlgeschlagen	Fehlermeldung nicht korrekt lokalisiert
24.05.2025	Omda	2	bestanden	UI-Elemente passen gut
24.05.2025	Maric	4	bestanden	–

5. Auswerten (Fazit & Reflexion)

Fazit:
Die .NET MAUI App erfüllt alle Kernanforderungen: einheitliche Codebasis, responsives UI-Design und zuverlässige Funktionen auf allen Zielplattformen. Der MVVM-Ansatz hat die Wartbarkeit deutlich verbessert.

Reflexion:

- **Stärken:**
  - Schnelle Prototypenentwicklung dank XAML und Hot Reload
  - Klare Trennung von UI und Logik durch MVVM
- **Schwächen:**
  - Erste Einarbeitung in MAUI verlief steil
  - Plattformunterschiede (z.B. File Picker) erforderten Extra-Brücken
- **Ausblick:**
  - Automatisierte UI-Tests integrieren
  - CI/CD-Pipeline für Multi-Target-Builds aufsetzen
  - Erweiterung um Synchronisation mit Cloud-Datenbank